The Dissertation Committee for Joon Sung Park
certifies that this is the approved version of the following dissertation:

System Level Methodology for Low Cost Performance Characterization of Analog and Mixed-Signal Circuits

Committee:

Jacob A. Abraham, Supervisor

Nur Touba

Ranjit Gharpurey

Arjang Hassibi

Xiaojing Zhang

System Level Methodology for Low Cost Performance Characterization of Analog and Mixed-Signal Circuits

by

Joon Sung Park, B.E., M.S.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

August 2009

Dedicated to my beloved family

Acknowledgments

At first, I would like to express sincere gratitude to my advisor, Dr. Jacob A. Abraham for his support and insightful advice. He always encourages me to look at the big picture, and at the same time look at the fundamentals of the problem. His encouragement helped me greatly in doing the right research successfully. With his enthusiasm and breadth of knowledge, I could enjoy the long journey of Ph.D. research. I also would like to thank to Dr. Nur Touba, Dr. Ranjit Gharpurey, Dr. Arjang Hassibi and Dr. John X.J. Zhang for their time and guidance while being members of my dissertation committee.

My gratitude extends to Hongjoon Shin, Byungho Kim, Jaeyong Chung, Jae-Wook Lee, Rajeshwary Tayade, Chaoming Zhang, Kihyuk Han, Ji Hwan Chun, Tung-Yeh Wu and Jiseon Park for their valuable discussions. Also, I would like to appreciate all CERC members and ECE friends, Yonghyun Kim, Joon-Sung Yang, Jungho Jo, Kyungeun Kwon, Wonsoo Kim, Taesoo Jun, Sriram Sambamurthy, Romi Datta, Ramtilak Vemu, Sankar Gurumurthy, Eun Jung Jang, Junyoung Park, Hyun Jin Kim, Whitney Townsend and Gaurav Nolkha, Melissa Campos, Debi Prather and Andrew Kieschnick. For last four years, I truly enjoyed the life in Austin with these friends and colleagues.

My three internships at Qualcomm and Texas Instruments provided me excellent opportunities to learn cutting-edge technologies and industrial

problems. I would like to thank to Chris Barr, Srinadh Madhavapeddi, Alex Paglieri, Kye-Shin Lee and Hochul Kim from Texas Instruments, and Paul Bassett and Baker Mohammad from Qualcomm for their mentoring and collaboration. Also, I would like to express my gratitude for Eonjo Byun and Sejang Oh from Samsung for their gracious support and helpful discussions.

Finally, my best gratitude goes to my parents and my sister for their sincere support. Without their continuous encouragement and believing, this journey would not have been a success.

System Level Methodology for Low Cost Performance Characterization of Analog and Mixed-Signal Circuits

Publication No. _____

Joon Sung Park, Ph.D.
The University of Texas at Austin, 2009

Supervisor: Jacob A. Abraham

Conventionally, the performances of Analog and Mixed-Signal (AMS) circuits have been characterized using specification-based functional tests. In these test methods, the correct functionalities of AMS circuits are verified by measuring pre-determined specification parameters of AMS circuits. The conventional test methods provide accurate test results by using various test equipments which generate functional test signals and capture the test responses externally. However, due to rapid increase in the performance of AMS circuits in recent years, the conventional test methods face various challenges in the aspects of test cost, test time and testability.

The goal of this dissertation is to develop innovative functional test methods for AMS circuits which are aimed at reducing the test cost and test time while providing comparable test accuracy to the conventional test methods. To achieve this goal, efforts have been made to explore the characteristics

of AMS circuits in a system level and to research efficient performance characterization methods based on the system level modeling of Devices Under Test (DUTs). As a part of these efforts, the pseudorandom test methods for nonlinear AMS circuits have been developed. In these methods, the pseudorandom signal is used to excite the DUT and to generate the test response which has sufficient information to characterize DUT performances. The pseudorandom test methods use the Volterra series model to capture the nonlinear behaviors of AMS circuits and to calculate various specification parameters of the DUT using the pseudorandom test response. In doing so, the performances of nonlinear AMS circuits can be characterized straightforwardly and accurately using a low-cost test setup. Also, in an effort to reduce the test time, parallel test methods of AMS circuits have been developed in which multiple DUTs are tested simultaneously by sharing a common test setup. In these methods, the test responses generated from different DUTs are combined together and the resulting composite test response is used to characterize the performance of each DUT individually. This will reduce the use of tester resources and will increase the test throughput beyond the level limited by the test equipments. The spectral characteristics of test stimulus are studied along with the system level behavior of AMS circuits to develop the efficient parallel test methods. Finally, in order to consider the practical issue of generating at-speed test stimuli for high-speed DUTs using a low-cost test setup, a reconfigurable built-off test interface is developed which can be used to generate various test patterns, including high-speed pseudorandom signal, using a low-speed tester.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

For the last several decades, Analog and Mixed-Signal (AMS) Integrated Circuit (ICs) performance has been improving rapidly, fueled by growing demands from applications such as wireless communications and multimedia. In addition, the advances in manufacturing technology has enabled various analog and mixed-signal function blocks to be integrated into a single System-on-Chip (SOC) along with digital circuits. Typically, these improvements in IC development come with increases in complexity and process shrinks. These trends, in turn, increase the susceptibility of chips to various types of faults, thus requiring more accurate tests than before. While the importance of testing high-performance devices has been increasing drastically, testing these devices currently requires a large amount of capital investment, which is especially true for AMS circuits.

Conventionally, most AMS circuits are tested based on a specification-based test approach. In this approach, the functionalities of AMS circuits are verified by measuring various specification parameters using DC or AC test inputs, and by comparing the values of measured parameters against pre-

determined specification limits. This is the most straightforward method to test the circuit performance in the sense that the test input signals are the same as the functional input signals. Consequently, various specification-based test methods have been used successfully in characterizing the performances of AMS circuits accurately until now [11].

However, this success comes at the expense of increasing test cost and test time. Speeds of the Devices Under Test (DUTs) have been increasing rapidly for the last couple of decades, and this trend requires Automatic Test Equipment (ATE) to be frequently upgraded to faster and more expensive versions, which incurs a large amount of capital expenditure. Also, ever-increasing resolution of the DUT increases the test time to maintain accurate test results by collecting more samples of test responses than before. In the current IC development, increasing the test time pushes the test cost higher by a large amount. As a result, nowadays, the test cost represents a significant portion of the overall production cost of AMS circuits [3]. Thus, reducing the test cost and test time becomes one of the most important issues in AMS IC development.

In addition to the issues of increased test cost and test time, the SOC trend poses another significant challenge for testing embedded AMS circuits. Due to the tight constraint on the number of I/O pins and the routing issues, SOC devices do not provide access to every input or output of embedded AMS circuits. Thus, it becomes difficult to control the input and output of the DUT separately from other modules implemented in the same SOC

device. As a result, the external test equipment has only limited access to the embedded AMS circuits which results in reduced test accuracy compared to the conventional module-based test method.

The issues explained above call for innovative test methods to decrease the test cost and test time while not compromising the test accuracy. Also, a new test method is required to increase the testability of the embedded AMS circuits without increasing the manufacturing cost and the test cost.

1.2 Organization and Contribution of the Dissertation

This dissertation presents an efficient test framework for analog and mixed-signal devices based on a system level modeling of the DUT. By abstracting the behaviors of AMS circuits to the system level, the test framework presented in this dissertation can be flexibly applied to various applications which have different underlying implementation details.

The presented test framework aims to achieve three major goals. The first goal is to use the framework to characterize the performance of AMS circuits at low-cost while maintaining comparable test accuracy to the conventional test methods. To achieve this goal, the proposed test method exploits the spectral performance characterization algorithm using easy-to-generate test signals which are substantially digital signals that can be generated by a simple digital hardware or design for test (DFT) circuitry. A significant portion of testing AMS circuits is transferred to the digital domain. Moreover, the method is aimed at testing embedded AMS circuits efficiently by using

on-chip digital modules which are commonly available in SOC devices.

The second goal is to increase the test throughput by testing multiple AMS circuits simultaneously, and thus to reduce the effective test time per unit DUT. When testing multiple numbers of DUTs in parallel, the major issue is to characterize the performance of each DUT separately. To solve this issue, efforts have been made to extract the performance parameters of each DUT separately from the composite test response by exploring the signal properties of the test stimuli and the test response. In doing so, the test time can be reduced noticeably without compromising the test accuracy.

The final goal is to reduce the test cost associated with the ATE. In practical test environments, most AMS circuits require at-speed test stimuli. This imposes a big burden on the ATE as the speed of AMS circuits are increasing rapidly, and thus increasing the test cost considerably. In order to address this issue and to implement the test framework in this dissertation cost-effectively, an off-chip test interface is developed to execute the at-speed test of the high-speed DUT using a low-speed ATE.

The rest of this dissertation is organized as follows. First, Chapter 2 reviews the previous methods to test analog and mixed-signal circuits to highlight the motivation of the work presented in this dissertation.

Chapter 3 presents the low-cost test method to characterize the non-linear performance of AMS circuits as well as those embedded in SOC devices. This method employs a Volterra series to model the behavior of AMS circuits

in a system level and uses a pseudorandom signal to identify the parameters of the Volterra series model. The major contributions of this research can be summarized as follows.

- Develop a new functional test method using a pseudorandom signal, which can be easily generated from a Linear Feedback Shift Register (LFSR), to identify the Volterra series and to characterize the nonlinear performance of the DUT. Since the LFSR can be easily found in the SOC devices for various purposes, the proposed method can be easily extended to test embedded AMS circuit accurately.

- Come up with a low-complexity algorithm to identify the parameters of the Volterra series model (which represents the DUT performance accurately), and verify the validity of the presented pseudorandom test algorithm using hardware measurement.

Chapter 4 discusses an efficient parallel test method for AMS circuits to reduce overall test time. In the parallel test method, multiple DUTs share a common test setup to reduce the use of expensive tester resources and to decrease the total test time by increasing the number of DUTs tested per unit time. The main contributions of this research include the following.

- Develop a loopback test method for mixed-signal circuits which is not affected by the fault masking problem. Simple analog circuits, an analog adder and a RMS detector, are placed in the analog path to resolve the

fault masking issue. The digital-in/digital-out configuration achieved by the presented loopback test method helps to reduce the test cost noticeably by using digital test equipments to test AMS circuits.

- Devise a parallel test method to characterize the performance parameters of multiple AMS circuits simultaneously without increasing the use of tester resources. The parallel test method uses either pseudorandom signals or sinusoidal signals to excite multiple DUTs at the same time and to generate the composite test response which is post-processed to provide performance parameters of each DUT separately. Parallelism is increased by sharing common test equipment and a DUT board among multiple DUTs.

Chapter 5 presents an efficient test framework to extend the use of low-cost ATE to the at-speed test of high-speed DUTs. In order to bridge the speed gap between the ATE and the DUT, an off-chip test interface circuit, called Built-off Test Interface (BOTI), has been developed, and used to run at-speed test procedures on behalf of the ATE. The contribution of this research can be summarized as follows.

- The proposed test framework enables the low-cost ATE to actively control the high-speed test procedure using the built-off test interface. The built-off test interface and the proposed test framework are highly flexible, and can be used in various test applications without modifications.

- Devise a method to perform a reliable off-chip signal communication, and verify its validity through actual measurements.

Conclusions are presented in Chapter 6 with discussions on future work.

Chapter 2

Overview of Analog and Mixed-Signal Circuit Test

This chapter first presents a brief overview of the conventional test approach for analog and mixed-signal circuits, and explains why it raises some issues for recent AMS IC developments. Then, the review of various existing test approaches aimed at solving the problems with the conventional test method is presented along with the discussions on the pros and cons of each approach.

2.1 Conventional Test Approach

Figure 2.1 depicts a typical test setup for an AMS circuit. An external signal generator is used to apply functional test input to a DUT. For AMS circuit test, an Arbitrary Waveform Generator (AWG) or a sine wave generator is used to generate an analog test input while a pattern generator or a clock generator is used to generate a digital test input. Typically, the specification-based test of AMS circuits requires at-speed test inputs which means that the external waveform generator should be able to generate the test input signal as fast as the operating frequency of the DUT.

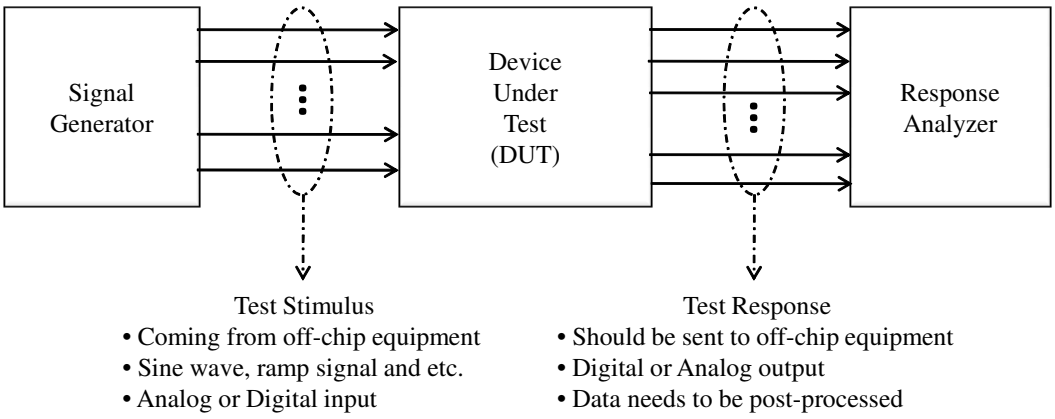


Figure 2.1: Conventional Test Setup

The output of the DUT is captured using a response analyzer and post-processed to verify the correct functionalities of the DUT. Various external test equipments, such as a spectrum analyzer, a digitizer or a network analyzer, can be used based on applications to capture the DUT output and to calculate the specification parameters. To meet the at-seed test requirements, these response analyzers also need to operate at least at the same speed as the operating frequency of the DUT. In case frequency domain characterization is required, the response analyzer should operate at least at twice the speed of the DUT to satisfy the Nyquist criterion.

Usually, the specification parameters of the AMS circuits are calculated in the frequency domain. This means that the DUT output should be converted into the frequency domain using a Fast Fourier Transform (FFT) operation. In this case, the accuracy of the test is greatly affected by the frequency resolution of the FFT, and this frequency resolution is determined

by the sampling frequency and by the number of samples. For a given time frame, the frequency resolution increases as the sampling frequency increases. Also, for a given sampling frequency, the frequency resolution increases as the number of samples increases.

Although this conventional test setup offers good test accuracy since the functionalities of the DUTs are tested directly, it requires a long test time which increases with the resolution of the DUTs. Moreover, the cost associated with the external test equipment grows with the increase in the speed and the complexity of the DUTs. Even worse, in a SOC environment, it is difficult to access embedded AMS circuits from the external test equipment. Therefore, testing the embedded DUTs using the conventional test setup is becoming increasingly difficult.

2.2 Signature-based Test Approach

A signature-based test approach is aimed at reducing the test cost by using an indirect method to characterize the performance of AMS circuits using a low-cost test setup. In this approach, a non-conventional test signal, which can be easily generated from on-chip built-in test circuitry or low-cost external test equipment, is applied to the DUT, and then the response (which is called a *signature*) of the DUT is analyzed to estimate the performance of the DUT indirectly.

Various studies have been conducted to use easy-to-generate signals such as DC values, piecewise linear or noise signals, to excite AMS circuits

and to produce the signature that has a good correlation with the actual DUT performance. Notable among them is the pseudorandom test which has been proven to be an efficient low-cost signature-based test algorithm for analog and mixed-signal circuits [7, 15, 22, 32, 35, 38]. In a typical pseudorandom test implementation, the input sequence generated from a LFSR is used to excite a DUT. Then, the cross-correlation of the input and output sequence is calculated and used as a signature to predict the performance parameters [35]. The major advantage of using a pseudorandom signal is that it is a spread-spectrum signal that covers a wide range of frequencies. In this case, the cross-correlation of the input and output represents the transfer function of the DUT [51]. It is well known that the transfer function determines the behavior of a certain system in both time and frequency domains. Thus examining the transfer function will detect changes in the system properties. Another advantage of the pseudorandom test is that it can be easily embedded into a SOC environment by re-using the LFSR which is widely used in the area of digital circuit Built-in Self Test (BIST). This reduces the overhead of the test stimulus generation circuitry and allows the analog circuit domain to be highly integrated with the digital circuit domain.

The aforementioned pseudorandom test has been very successful in detecting both catastrophic and parametric faults of the linear circuits that cause changes in the transfer function. However, these methods cannot capture the nonlinear behavior of the DUT since the primary assumption of the conventional pseudorandom test is that the DUT should be represented as a Lin-

ear and Time-Invariant (LTI) system [51]. It has been considered difficult to identify a nonlinear system function using pseudorandom sequences. The main difficulty comes from the fact that nonlinear behavior modeling requires advanced mathematical methods which increase the complexity of the performance characterization method [56]. There have been several studies to apply pseudorandom test schemes to mixed signal [15, 32] and RF [7] nonlinear circuits without deriving a nonlinear system function directly. In these studies, performance parameters were predicted by mapping the measurement space to the parameter space. A Chebychev polynomial was used in [15] to bridge the nonlinear static errors (integrated nonlinearity and differential nonlinearity) of Analog-to-Digital Converters (ADCs) and the output signatures, while a nonlinear regression technique was used in [7] to map the performance parameters of RF components to the alternate test outputs. These methods could predict the performance of the nonlinear circuits with a small prediction error. However, they require additional input stimuli other than the pseudorandom signal to generate the output signature, and the pseudorandom signal was mainly used to sample the output signature of the DUT. This may increase the amount of test pattern generation circuitry and also raise the test cost.

Another direction in the signature-based test approach is to predict the DUT performance by mapping the information gathered from the signature of the DUT to the specification parameters of the DUT using statistical mapping functions [6, 7, 18, 24, 61] or mathematical equations [15, 48]. The major goal in this approach is to replace the costly conventional methods of functional

test with an alternate test method that can be executed using a low-cost test setup. In doing so, the test cost can be considerably reduced compared to the conventional test methods while the test accuracy can be improved compared to the typical signature-based tests explained above. The underlying principle in this approach is to use the correlation between the low-cost test result (measured signature) and the conventional test result (specification parameter). This correlation arises from the fact that both test results are sensitive to same fault mechanisms and process variations of the DUT which lead to performance deviations. Thus, the objective of this approach is to exploit the cost-effective test method to generate a signature that is highly correlated with the DUT performances and to use the proper mapping tool to predict the specification parameters from the measured signature.

To achieve this objective, on one hand, there have been a number of studies aim at finding proper low-cost test stimuli that offer good correlation between the measured signature and the specification parameters. For example, multi-tone sinusoidal signals [31], piecewise linear signals [53] and noise signals [7, 15] were used in different cases to test various analog and RF circuits. Moreover, the work presented in [8, 43] eliminates the use of a test stimulus by converting the DUT into an oscillator.

On the other hand, several studies focused on generating the low-cost representation of the DUT output when a conventional test input is applied to the DUT. The work presented in [18, 61] used a built-in on-chip sensor to sense the amplitude of the RF signal and to predict the performance of the RF

circuits at high frequencies. This method makes it possible to predict the high-frequency performance of the RF circuit using a low-frequency test equipment. Moreover, in [25, 60], the authors used a simple analog comparator to represent the sinusoidal signal with a three-level DC signal, and thus to reduce the cost associated with the response analyzer.

The alternate test methods described above have been used successfully in many applications to reduce the test cost and test time. The main issue with this approach, however, is that the test accuracy is limited by the accuracy of the mapping function, while there is no metric to estimate the accuracy of the mapping function quantitatively. Typically, extensive circuit simulations are required to ensure that the derived mapping function represents the correlation between the measured signature and the specification parameters accurately. However, it is very difficult to derive an accurate mapping function when there is a high degree of nonlinearity between the measured signature and the specification parameters, or when the netlist of the DUT is not available.

2.3 On-chip and Off-chip Test Dedicated Circuitry

As explained previously, one of the important factors which increases the test cost is the cost associated with the ATE. This is especially true for the at-speed test of high-speed devices, since the ATE needs to be frequently upgraded to faster and more expensive versions to support ever-increasing DUT speeds [33]. There have been a number of research projects in recent years attempting to relieve the ATE-related cost by performing various tests

using *test-dedicated circuitry* which could be implemented on the DUT (*on-chip*) or off the DUT (*off-chip*) [10, 14, 17, 30, 35, 42]. The common goal in all these research projects is to implement the ATE functions, such as test pattern generation or test response collection, using the test-dedicated circuitry so that the ATE does not have to handle high-speed signals externally. In this way, a low-cost ATE with slow speed and low I/O bandwidth can be used to test high-performance DUTs.

Various techniques have been proposed to implement ATE functions such as analog waveform generation, digitization and signal processing on-chip [17]. These function blocks can be implemented by reusing existing on-chip modules or designing separate built-in test circuitry. For example, analog waveforms can be generated on-chip using a Digital-to-Analog Converter (DAC) or an oscillator, and digitization can be implemented using an ADC, while signal processing can be done using an on-chip digital signal processor (DSP). By using on-chip test-dedicated circuits, we can replace analog test inputs or outputs with digital signals. Considering that digital signals are easier to generate and to replicate than analog signals, these techniques can help reduce the test cost [30]. For example, the work presented in [14] developed an on-chip signal generator for frequency-domain testing of ADCs. It used a static RAM to generate a digital sine wave and used a sigma-delta generator to convert the digital sine wave into an analog signal. In [42], the authors presented an on-chip ramp generation scheme for time-domain testing of analog circuits and histogram test of ADCs. Their design requires only a

system clock and a voltage reference as inputs to a ramp generator; therefore, a complex analog waveform generator is not required. Pan *et al.* [35] proposed a BIST scheme for testing linear analog circuits in which test input generation and response analysis are all performed on-chip. LFSR and DAC are used to generate the test stimulus, and on-chip DSP and ADC are used to analyze the output response.

The biggest challenge in these methods is to provide accurate test results with a reasonable amount of area overhead for test-dedicated circuits. Typically, the test signals (such as sine wave or ramp signal) generated on-chip are not as clean as the signals generated by external test equipment and this leads to inaccurate test results compared to conventional test methods.

Another approach is to implement test-dedicated circuits off-chip and use them to interface the low-cost ATE and high-performance DUT [10]. Since the off-chip method does not require extra on-chip test circuits, it can be used in various test applications, which normally require an external high-speed ATE for testing the DUT, without any manipulation of the DUT. However, in this method, the test-dedicated circuit has to communicate the signals with the DUT through an off-chip environment which makes the signal communication vulnerable to off-chip channel skew and hazardous glitches. This factor may lead to a decrease in the reliability of test results.

2.4 Parallel Test Method

One way to reduce the test time is to increase the test throughput by testing multiple DUTs in parallel [9, 21, 26, 46, 54]. This method of *parallel testing* or *multi-site testing* has been widely used in memory and digital test areas and has recently gained popularity in the AMS circuit test area [9, 58]. However, due to the limited I/O pin counts and the complexity of analog test components inside a mixed-signal tester which is used to test AMS circuits, the number of mixed-signal devices that can be tested simultaneously has been restricted to a small number. The ITRS (International Technology Roadmap for Semiconductors) report indicates that memory testers can support up to 512 parallel tests, while the mixed-signal testers can support up to 8 by the year 2007 [2, 3].

In order to increase the degree of parallelism beyond the level limited by the tester, various parallel test techniques for AMS circuits have been proposed recently. The authors in [46] used both time-domain and frequency-domain multiplexing algorithms to share one digitizer among multiple stereo DACs. They implemented a multiplexing logic on a DUT board and showed that the test time as well as the test cost can be reduced by sharing the common digitizer. Recently, Kwan *et al.* [26] proposed an algorithm to test two current steering DACs used in an RF CODEC with one digitizer. The outputs of these DACs are combined using resistor loads and are sent to a single digitizer to characterize overall performances. Also, Jin *et al.* [21] presented a technique to test high-resolution DACs on-chip using flash ADCs. In their algorithm, high

speed on-chip data acquisition and digitization is achieved using the ADCs, and thus external equipment is not required. The authors in [23] presented an algorithm to test multiple data converters under a digital test environment which has advantages over an analog test environment in the aspect of cost. In this method, the outputs of the DACs are connected to the inputs of the ADCs to configure a digital-in/digital-out test environment.

These techniques, however, are vulnerable to fault masking which may lead to considerable a yield loss and low test accuracy [47]. Fault masking occurs when the test response from one DUT is corrupted by interactions from other modules which share the same signal path with the DUT. This leads to misinterpretation of the test response observed at the output node, thus causing a pass/fail decision based on this observation to be incorrect.

Chapter 3

Pseudorandom Test of Nonlinear Devices

This chapter discusses pseudorandom test methods for analog and mixed-signal circuits which are developed based on a system level modeling of the DUT. The goal of studies presented in this chapter is to develop low-cost test algorithms which use a non-conventional test setup to characterize the performance of the nonlinear AMS circuits accurately, without using any indirect mapping methods. To achieve this goal, the presented algorithms use a pseudorandom sequence, which can be easily generated from a built-in test circuitry such as a LFSR or a low-cost test equipment, to find the fundamental description of the DUT behavior, and to predict the performance of the DUT.

There are two pseudorandom test methods presented in this chapter. In both methods, a Volterra series is used to model the DUT at the system level. The first method, which presented in Section 3.2, uses a *simplified* Volterra series model to reduce the complexity of the test algorithm. On the other hand, the second method, which presented in Section 3.3, uses a general discrete-time Volterra series model while a *compressed cross-correlation* method is used to reduce the complexity. As will be described later in this chapter, the first method can be used in applications in which memory effects of the circuit can

be ignored while the second method can be used in more general cases.

3.1 Review of Pseudorandom Test Method

This section presents a brief review of the existing pseudorandom test method detailing its fundamental idea as well as its limitations.

3.1.1 Conventional Pseudorandom Test

It is well known that a LTI system can be described by the transfer function as follows,

$$y(t) = x(t) * h(t) \tag{3.1}$$

where $h(t)$ is the transfer function of DUT. The basic idea of pseudorandom test is to find the transfer function or some other parameters closely related to the transfer function. By comparing the key parameters of the obtained transfer function with the golden values, we can see whether the performance parameters of the DUT stay within the tolerance range. A random sequence $x(t)$ generated by a specific random process X is used to excite the DUT. When $x(t)$ passes through the DUT, a new random process Y, which generates the random sequence $y(t)$, is formed at the output. To find the transfer function of the DUT, we first need to find the cross-correlation of the random variable

X and Y which is given as follows.

$$\begin{aligned} R_{yx}(m) &= E[(x(t) * h(t)) \cdot x(t-m)] \\ &= E[\sum_{i=0}^{\infty}(x(t-i)h(i))x(t-m)] \\ &= \sum_{i=0}^{\infty}(E[x(t-i)x(t-m)]h(i)) \\ &= R_{xx}(m) * h(m) \end{aligned} \tag{3.2}$$

where $R_{xx}(m)$ is the auto-correlation function of DUT. By taking the Fourier transform, we can obtain the cross spectral density.

$$S_{yx}(\omega) = H(\omega)S_{xx}(\omega) \tag{3.3}$$

In Equation 3.3, the power spectral density of X, $S_{xx}(\omega)$, can be calculated *a-priori*, and the cross spectral density between X and Y, $S_{yx}(\omega)$, can also be calculated by post-processing the output random sequence Y with X. Then we can find the transfer function $H(\omega)$ using Equation 3.3.

3.1.2 Issues in Applying Pseudorandom Test to Nonlinear Circuits

For non-LTI systems, the output random process Y is not a linear transform of the input random process X [51]. In this case, the cross correlation between the input and output corresponds to the *first-order kernel* which represents the linear operation of the circuits. This means that the conventional pseudorandom test cannot be applied to nonlinear circuits where the *second and higher order kernels* play important roles in characterizing the nonlinearity. Even in linear circuits, nonlinearity cannot be avoided due to

various types of parasitics. This means that the conventional pseudorandom test scheme may not be able to provide enough information about the DUT in practical situations. Therefore the conventional pseudorandom test scheme needs to be improved to capture the nonlinearity of the DUT.

3.2 Review of the Volterra Series Model

The work presented in this chapter uses a Volterra series to model nonlinear behaviors of DUTs. The Volterra series describes the output of a nonlinear system as a sum of response of a first-order operator, a second-order one and so on [57]. A general Volterra series can be represented by multi-dimensional convolutions as follows.

$$y(t) = \sum_{k=1}^{N} \sum_{\tau_1=0}^{L} \cdots \sum_{\tau_k=0}^{L} h_k(\tau_1, \cdots, \tau_k) x(t-\tau_1) \cdots x(t-\tau_k) \qquad (3.4)$$

where x and y are input and output, respectively, and h_k represents the kth-order Volterra kernel. Using the Volterra series model, the kth-order nonlinear behavior can be described by the kth-order Volterra kernel.

It has been shown that a wide class of nonlinear systems can be represented as a Volterra series of finite order N and finite memory L [57]. Especially for most analog circuits which exhibit weakly nonlinear behavior, the impact of the fourth and higher order nonlinearities on the performance modeling is negligible [45]. Thus, in this chapter, we consider the Volterra series up to the third order, and assume that the Volterra kernels are discrete in time with finite memory length L. Moreover, it is assumed that the Volterra

kernels are symmetric, which means that the Volterra kernel of each order, $h_k(\tau_1, \cdots, \tau_k)$, has same value regardless of permutation of indices, $\tau_1, \cdots, \tau_k$. This is common assumption when modeling nonlinear behavior of AMS circuits using Volterra series [57].

Once the system model has been decided on, the next step is to come up with an identification algorithm to find the values of model parameters. In the case of the test methods presented in this chapter, the model parameters refer to the values of Volterra kernels, and from a testing perspective, the goal is to identify the Volterra kernels using the proper test stimulus applied to the DUT and analyzing the resulting DUT outputs. Conventionally, the values of Volterra kernels can be identified using either sinusoidal signals or random (or pseudorandom) signals as input to the system. In the presented work, a pseudorandom signal is used as a test stimulus based on two reasons. First, the pseudorandom signal can be easily generated using a LFSR which is commonly used in SOCs for various testing purposes [58]. Secondly, the pseudorandom signal is essentially a spread-spectrum signal, and thus it covers a wide range of frequencies. Therefore, using the pseudorandom signal, performance parameters of the DUT in various frequency ranges can be characterized without frequency sweeping, and this will save the test time considerably. When using a pseudorandom signal as the test input, conventionally, a cross-correlation method is used to identify the values of the Volterra kernels. In this method, each component of the kth-order Volterra kernel can be identified using $(k-1)$th-order cross-correlation between input and output [27].

The issue with the conventional cross-correlation method is that the computation complexity grows exponentially with the order of the Volterra series. The kth-order kernel with memory length L has $\binom{L+k-1}{k}$ components, and to identify them all, it requires $\binom{L+k-1}{k}$ cross-correlation calculations which increase with the order, k, and the memory length, L. Moreover, the accuracy of the cross-correlation method depends on the statistics of the input signal which is supposed to have a Gaussian distribution. However, this requirement is difficult to meet using the test stimulus generated from a simple LFSR. Due to these issues, it is impractical to adopt the conventional identification method directly in testing nonlinear devices.

3.3 Pseudorandom Test using a Multi-level Pseudorandom Sequence

This section presents an efficient pseudorandom test algorithm for nonlinear circuits based on a simplified Volterra series model. The simplified Volterra series model is employed to represent the nonlinear behavior of the DUT in a linear fashion and to make it possible to reduce the complexity of transfer functions up to the third order. Then the parameters of the Volterra series (Volterra kernels) are identified using Pseudorandom Multi-level Sequences (PRMS). The PRMS have been used in the signal processing area to find the Volterra kernels in a computationally efficient way [34]. A simplified version of the PRMS is used in this research to make the test stimuli generation scheme practical. Derived Volterra kernels are then used to predict

the performance parameters of the DUT and also used to compensate for the nonlinear errors of the DUT.

3.3.1 Volterra Kernel Identification based on a Simplified Volterra Series Model

In order to reduce the complexity of the Volterra kernel identification, we will assume that the only convolutions taken are those associated with the power of the input sequence. Then the Volterra series can be written as follows.

$$y(t) = h_1(t) * x(t) + h_2(t) * x^2(t) + h_3(t) * x^3(t) \tag{3.5}$$

where h_1, h_2 and h_3 correspond to the first, second and third order Volterra kernels, respectively. For a memoryless nonlinear system, this gives an accurate representation within the third order. As memory is introduced, there is a degree of approximation introduced, although the reduction in computational complexity is a desirable outcome [19]. The rest of this section presents the Volterra kernel identification scheme based on the simplified Volterra series model described above.

As in a conventional pseudorandom test, the input stimulus is a sequence of pseudorandom patterns $x(t)$ generated from the pseudorandom signal generator which consists of an LFSR and a Digital-to-Analog Converter [35]. $x(t)$ is set to have mean zero and variance σ_x^2. Now, the input and output relationship of the nonlinear DUT can be represented in a Volterra series as

follows.

$$y(t) = h_1(t) * x(t) + h_2(t) * x^2(t) + h_3 * x^3(t) + N(t) \tag{3.6}$$

where $N(t)$ represents the output referred noise. In this section, it is assumed that the noise process is *independent and identically distributed* (i.i.d.) with mean zero and variance σ_n^2. This is a reasonable assumption considering that the output referred noise is usually modeled as white noise process which has mean of zero by definition [45, 51]. Now, we can write the cross-correlation of the input random sequence and the output random sequence as follows.

$$\begin{aligned} R_{yx}(m) &= E[y(t)x(t-m)] \\ &= h_1(m) * R_{xx}(m) + h_2(m) * R_{x^2x}(m) \\ &+ h_3(m) * R_{x^3x}(m) + E[N(t)x(t-m)] \end{aligned} \tag{3.7}$$

where $R_{xx}(m)$ is the auto-correlation of $x(t)$ while $R_{x^2x}(m)$ and $R_{x^3x}(m)$ are cross-correlations of $x^2(t)$, $x(t)$ and $x^3(t)$, $x(t)$, respectively. The last term in Equation 3.7 becomes zero since $N(t)$ and $x(t)$ are uncorrelated and $x(t)$ has mean zero as shown in the following equation.

$$E[N(t)x(t-m)] = E[N(t)]E[x(t-m)] = 0 \tag{3.8}$$

Now, let us change the problem solving domain from the time region to the frequency region, since this allows us to make use of well-known linear algebra techniques in finding the Volterra kernels. The Fourier transform of $R_{yx}(m)$ is the cross spectral density of $x(t)$ and $y(t)$ which can be represented as follows.

$$\begin{aligned} S_{yx}(\omega) = \mathcal{F}(R_{yx}(m)) &= H_1(\omega)S_{xx}(\omega) \\ &+ H_2(\omega)S_{x^2x}(\omega) + H_3(\omega)S_{x^3x}(\omega) \end{aligned} \tag{3.9}$$

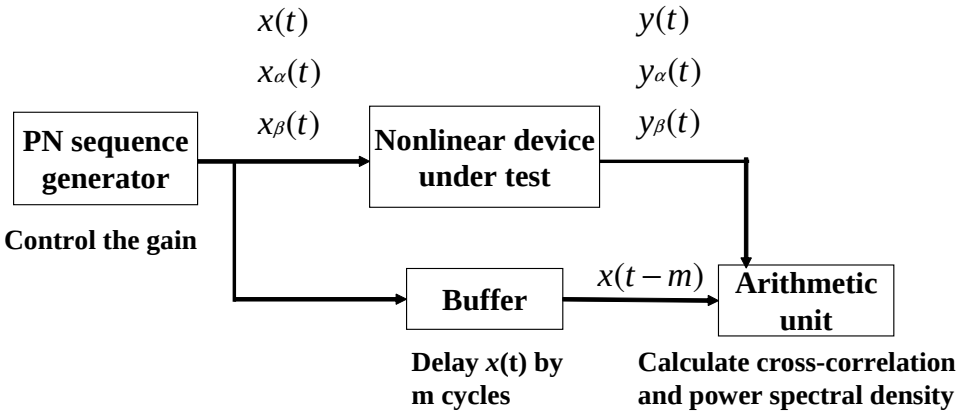


Figure 3.1: Proposed Pseudorandom Test Scheme

where $H_1(\omega)$, $H_2(\omega)$ and $H_3(\omega)$ correspond to frequency domain Volterra kernels. $S_{xx}(\omega)$ is the power spectral density of $x(t)$, and $S_{x^2x}(\omega)$, $S_{x^3x}(\omega)$ are the cross spectral densities between $x^2(t)$, $x(t)$ and $x^3(t)$, $x(t)$ respectively. These spectral density values can be easily calculated using the input and output sequences. Then we have three unknown terms left in Equation 3.9. Note that these unknown terms correspond to the Volterra kernels we need to find. This tells us that two more equations linearly independent to Equation 3.9 are required to identify all three Volterra kernels, $H_1(\omega)$, $H_2(\omega)$ and $H_3(\omega)$. The concept of the PRMS is used here to form these linearly independent equations [34]. At first, one set of pseudorandom patterns is generated and then two more sets of pseudorandom patterns are generated by applying constant gains to the first set of pseudorandom patterns. Figure 3.1 shows the proposed test scheme using the PRMS. Assume that the gain is changed by factors of α and β ($\alpha \neq \beta$). Then the input and output relationships of the DUT can be

represented as follows.

$$\begin{aligned} y_\alpha(t) &= h_3(t) * (\alpha x(t))^3 + h_2(t) * (\alpha x(t))^2 \\ &\quad + h_1(t) * (\alpha x(t)) + N_\alpha(t) \\ y_\beta(t) &= h_3(t) * (\beta x(t))^3 + h_2(t) * (\beta x(t))^2 \\ &\quad + h_1(t) * (\beta x(t)) + N_\beta(t) \end{aligned} \tag{3.10}$$

where $y_\alpha(t)$ and $y_\beta(t)$ are outputs of the DUT in response to the inputs $x_\alpha(t)$ and $x_\beta(t)$ respectively. We can find the cross spectral densities between $y_\alpha(t)$, $x(t)$ and $y_\beta(t)$, $x(t)$ using Equations 3.7 and 3.9. After some calculation, $S_{y_\alpha x}(\omega)$ and $S_{y_\beta x}(\omega)$ can be represented as follows.

$$\begin{aligned} S_{y_\alpha x}(\omega) &= \alpha^2 H_1(\omega) \cdot S_{xx}(\omega) + \alpha^3 H_2(\omega) \cdot S_{x^2x}(\omega) \\ &\quad + \alpha^4 H_3(\omega) \cdot S_{x^3x}(\omega) \\ S_{y_\beta x}(\omega) &= \beta^2 H_1(\omega) \cdot S_{xx}(\omega) + \beta^3 H_2(\omega) \cdot S_{x^2x}(\omega) \\ &\quad + \beta^4 H_3(\omega) \cdot S_{x^3x}(\omega) \end{aligned} \tag{3.11}$$

We need to make sure that Equations 3.9 and 3.11 are linearly independent. To check this property, let us write these equations in a matrix form, $\mathbf{S} = \mathbf{TH}$, which can represented as follows.

$$\begin{bmatrix} S_{yx} \\ S_{y_\alpha x} \\ S_{y_\beta x} \end{bmatrix} = \begin{bmatrix} S_{xx} & S_{x^2x} & S_{x^3x} \\ \alpha^2 S_{xx} & \alpha^3 S_{x^2x} & \alpha^4 S_{x^3x} \\ \beta^2 S_{xx} & \beta^3 S_{x^2x} & \beta^4 S_{x^3x} \end{bmatrix} \begin{bmatrix} H_1 \\ H_2 \\ H_3 \end{bmatrix} \tag{3.12}$$

where the transform matrix ($\mathbf{T}$) can be decomposed as follows.

$$\mathbf{T} = \begin{bmatrix} 1 & 1 & 1 \\ \alpha^2 & \alpha^3 & \alpha^4 \\ \beta^2 & \beta^3 & \beta^4 \end{bmatrix} \begin{bmatrix} S_{xx} & 0 & 0 \\ 0 & S_{x^2x} & 0 \\ 0 & 0 & S_{x^3x} \end{bmatrix} \tag{3.13}$$

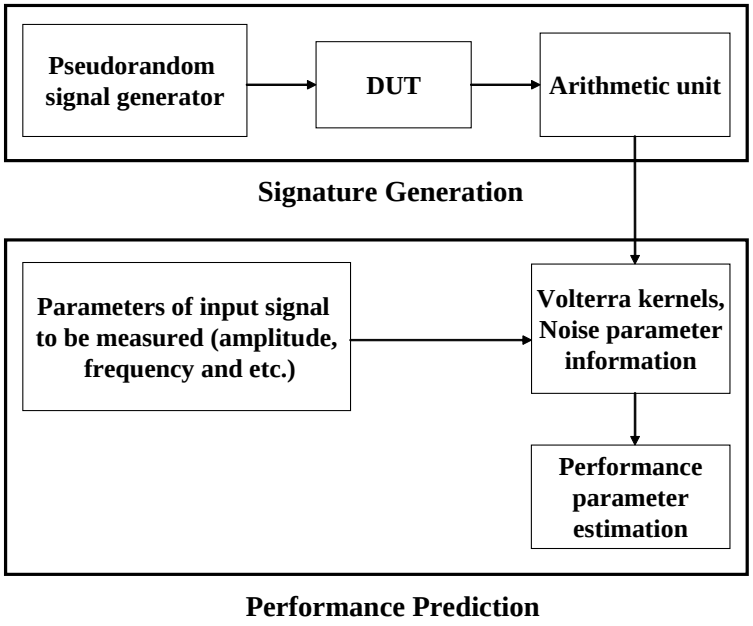


Figure 3.2: Performance Prediction Scheme

In Equation 3.13, the first matrix is invertible since it is in the form of a generalized Vandermonde matrix [16]. The second one is a diagonal matrix with nonzero elements across the diagonal, so this one is also invertible [51]. Thus the transform matrix $\mathbf{T}$ is a non-singular and invertible matrix. This means that the three equations that form the matrix equation in Equation 3.12 are linearly independent. Now that we know the transform matrix is invertible, the Volterra kernels can be identified using the following equation.

$$\mathbf{H} = \mathbf{T}^{-1}\mathbf{S} \tag{3.14}$$

3.3.2 Noise Power Calculation

This section presents an efficient method to find the power of the noise process N using pseudorandom sequences. In order to find the power of N, two sets of identical pseudorandom sequence with the same length are applied to the DUT. Then output of the DUT in each case can be represented as follows.

$$\begin{aligned} y_1(t) &= f(x(t)) + N_1(t) \\ y_2(t) &= f(x(t)) + N_2(t) \end{aligned} \tag{3.15}$$

where the function f represents the nonlinear function of the DUT and $N_1(t)$ and $N_2(t)$ are i.i.d random sequences that represent output referred noise in each case. Note that the two random sequences $N_1(t)$ and $N_2(t)$ have the same probability distribution since they are generated from the same noise process N [51]. Now, by subtracting $y_1(t)$ from $y_2(t)$, we get the following result.

$$y_2(t) - y_1(t) = N_2(t) - N_1(t) = N_s(t) \tag{3.16}$$

and the second moment of the $N_s(t)$ is given as follows.

$$\begin{aligned} E[N_s^2(t)] &= E[(N_2(t) - N_1(t))^2] \\ &= E[N_2^2(t)] + E[N_1^2(t)] - E[N_2(t)N_1(t)] \end{aligned} \tag{3.17}$$

In Equation 3.17, $E[N_2(t)N_1(t)]$ can be decomposed into $E[N_2(t)]E[N_1(t)]$ since $N_1(t)$ and $N_2(t)$ are independent to each other, and thus uncorrelated to each other. Now this term becomes zero since we already assume that the mean of the noise process is zero, *i.e.,* $E[N_2(t)] = E[N_2(t)] = 0$. Then, Equation 3.17 can be simplified to the following equation.

$$E[N_s^2(t)] = E[N_2^2(t)] + E[N_1^2(t)] = 2\sigma_n^2 \tag{3.18}$$

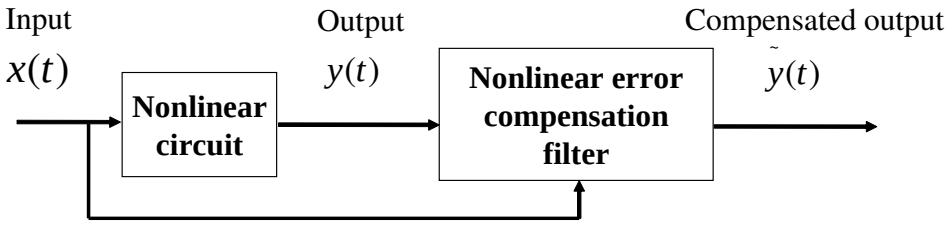


Figure 3.3: Nonlinear Error Compensation Scheme

So, we can find the power of the output referred noise, σ_n^2 as

$$\sigma_n^2 = 0.5E[(y_2(t) - y_1(t))^2] \tag{3.19}$$

3.3.3 Performance Prediction and Nonlinear Error Compensation

Once we find the information of the Volterra kernels and noise parameter as discussed in the Sections 3.3.1 and 3.3.2, we can use this information to predict the performance of the DUT in response to various input signals. Figure 3.2 describes the concept of using Volterra kernels and noise parameter information to predict the performance parameters of the DUT. Using this concept, we can examine the performance of the DUT for various input conditions without actually applying these inputs to the DUT. This will lead us to characterize the behavior of the DUT accurately in a short period of time. In addition to the parameter prediction, we can use the information of the Volterra kernels to compensate for the performance deviation introduced by the nonlinearity of the DUT [29]. The basic concept is to cancel out the unwanted output response generated by the second-order and third-order

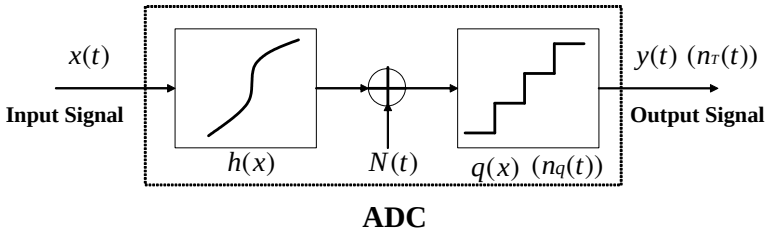


Figure 3.4: Nonlinear ADC model for simulation

Volterra kernels. Figure 3.3 describes the conceptual view of the nonlinear error compensation scheme. The following computation is done at the nonlinear error compensation filter.

$$\tilde{y}(t) = y(t) - h_2(t) * x^2(t) - h_3(t) * x^3(t) \tag{3.20}$$

where $h_2(t)$ and $h_3(t)$ are the second and third order Volterra kernels.

3.3.4 Simulation Results

The pseudorandom test method described until now was applied to a 16-bit *Sample and Hold Analog-to-Digital Converter* (S/H ADC) with MATLAB simulation. The ADC is modeled as shown in Figure 3.4 where the ADC model is divided into two blocks: the first block models the dynamic nonlinearity of the ADC which is represented as $h(x)$, and the second block models the ideal quantization process which is represented as $q(x)$. Also, white Gaussian noise, $N(t)$, is added to the output of the nonlinear circuit. Then the output noise $n_T(t)$ consists of the Gaussian noise, $N(t)$, and the quantization noise, $n_q(t)$, generated during the quantization process.

This section consists of two parts. In the first part, the proposed method is applied to find the Volterra kernels and noise parameter of the ADC, and used this information to predict the performance parameters. In the second part, the characterized information is used in the first part to compensate for the nonlinear error of the ADC.

3.3.4.1 Performance Parameter Estimation

For simulation, 100 ensembles of the ADC model was generated by introducing statistical variations with a Gaussian distribution in the parameters of the nonlinear function, $h(x)$, described in Figure 3.4. As mentioned earlier, the presented test method requires pseudorandom sequences with three different gains. Three Gaussian random sequences are generated to have zero mean and standard deviations of $0.01V_{ref}$, $0.02V_{ref}$ and $0.04V_{ref}$ each, where V_{ref} is reference voltage of the ADC. 1000 samples of each sequence were taken at the output of the ADC, and used to identify the Volterra kernels and noise power. This information is used to predict values of *Total Harmonic Distortion* (THD), *Signal-to-Noise Ratio* (SNR) and *Signal-to-Noise and Distortion Ratio* (SNDR). Figure 3.5 shows the plots of the predicted versus the actual values of the ADC performance parameters. Table 3.1 summarizes the mean error of performance parameter prediction and correlation coefficients between the predicted values and the actual values. Next, the information of the Volterra kernels and the noise parameter are used to predict the performance of the ADC in response to various input signals and compared the predicted values

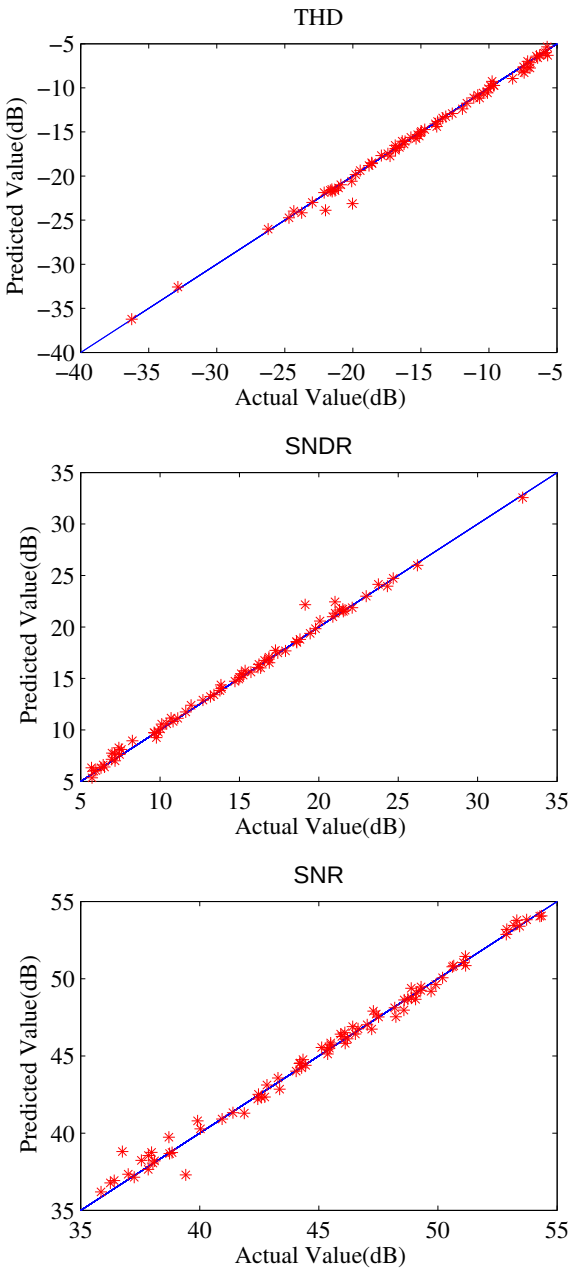


Figure 3.5: Comparison of actual and predicted values of THD, SNDR and SNR

Table 3.1: **Mean Error and Correlation Coefficient of Performance Parameter Prediction**

Performance Parameter	Mean Error	Correlation Coefficient
THD	0.50dB	0.9951
SNDR	0.50dB	0.9950
SNR	0.51dB	0.9940

Table 3.2: **Parameter Prediction Results for Various Input Conditions**

Input Condition		Mean Prediction Error		
Frequency	**Amplitude**	**THD**	**SNDR**	**SNR**
$0.01f_s$	-10dBFS	0.64dB	0.59dB	0.36dB
$0.001f_s$	-10dBFS	0.54dB	0.53dB	0.43dB
$0.1f_s$	-20dBFS	0.66dB	0.27dB	0.27dB
$0.01f_s$	-20dBFS	0.55dB	0.32dB	0.32dB
$0.001f_s$	-20dBFS	0.64dB	0.28dB	0.28dB

with the actual values. Table 3.2 summarizes the mean error of parameter prediction. We can see that the presented pseudorandom test method predicted the performance parameters of the ADC accurately for various input amplitudes and frequencies without physically applying these inputs to the ADC.

3.3.4.2 Nonlinear Error Compensation

As discussed in Section 3.3.3, we can compensate for the nonlinear error of the DUT by using the information of Volterra kernels. In this section, compensation effectiveness is demonstrated on a sinusoidal test signal of amplitude -10dBFS and frequency $0.1f_s$ where f_s is the sampling frequency of the ADC. Figure 3.8 (a) and (b) show the details of the effect of the nonlinear

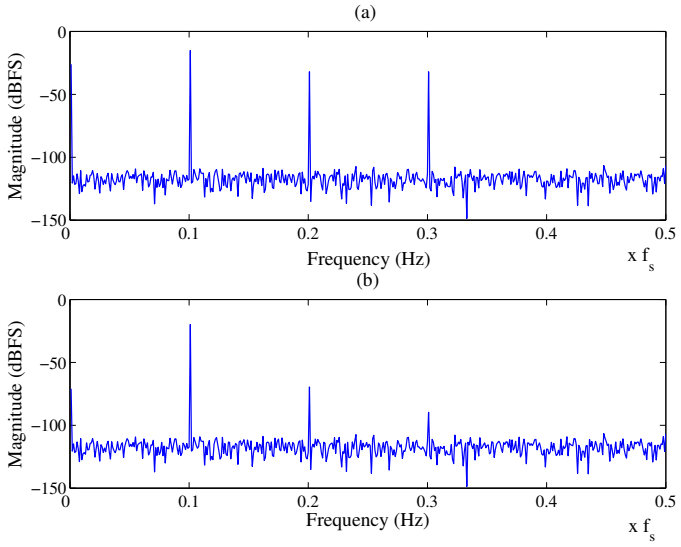


Figure 3.6: Details of Simulation Results in Frequency Domain: (a) Before Compensation (b) After Compensation

Table 3.3: **Performance Parameters of ADC: Before and After the Nonlinear Error Compensation**

Performance Parameter	**Before**	**After**
THD	-13.87dB	-49.66dB
SNDR	13.87dB	49.61dB

error compensation. It can be clearly seen that the second and third harmonic components were reduced substantially after the compensation. This compensation leads to a performance improvement of the DUT. In case of the ADC, this improvement can be examined by looking at the values of THD and SNDR which are summarized in Table 3.3.

3.3.5 Summary

In Section 3.3, an efficient pseudorandom test methodology for nonlinear circuits has been presented. The method described in this section uses a simplified Volterra series model to characterize the nonlinear behavior of a DUT. A sequence of multi-level pseudorandom signals is used as a test stimulus to excite the nonlinear DUT over a wide range of frequencies. The cross spectral density of the input and output are calculated, and used to identify the Volterra kernels. The identified Volterra kernels are used to predict the performance of the DUT and to compensate for the nonlinear errors. The mathematical background and simulation results exhibit the validity of the presented pseudorandom test method.

3.4 Pseudorandom Test using Higher-order Statistics

The previous section presented a pseudorandom test method to test nonlinear devices using a simplified Volterra series model. While that method extends the application of the pseudorandom test method to nonlinear circuits while retaining the computation complexity in the linear region, it requires multiple sets of the test stimulus with different amplitudes which can possibly increase the test time in adjusting the amplitude to different values. Moreover, in an effort to reduce the complexity, the method presented in the previous section does not consider the memory effects of the nonlinear devices which may lead to an insufficient characterization of AMS circuits.

This section presents a low-complexity algorithm to identify the param-

eters of the general discrete-time Volterra series model using a *single-amplitude* pseudorandom sequence. The theory of higher-order statistics is used in developing the proposed method to alleviate the complexity of the algorithm while maintaining comparable accuracy to the conventional performance characterization methods. In the method explained in this section, all the calculations required to identify the Volterra model are done in the time domain using a reduced-complexity cross-correlation method which results in a decrease in the test time. Moreover, in order to maintain good test accuracy, non-idealities of pseudorandom test input have been analyzed and their effects are reflected in the presented performance characterization algorithm. The validity of the algorithm is verified using hardware measurement of an actual mixed-signal circuit.

3.4.1 Review of Higher-order Statistics

Here, the basics of Higher-order Statistics (HOS) and higher-order moments which form the basis of the algorithm developed in this research are reviewed. The HOS measures are extensions of second-order measures to higher orders. In the HOS, statistics of a kth-order system can be represented using a kth-order cumulant which is defined by the cumulant-generating function as follows [28].

$$C(\boldsymbol{\tau}) = \ln E\{\exp(j\boldsymbol{\tau}'\boldsymbol{x})\} \quad (\boldsymbol{x} \text{ denotes a random process}) \tag{3.21}$$

For a *zero-mean* random process, the cumulants up to the third-order can be simplified as follows.

$$\begin{aligned} C_{1,x}(\tau) &= E[x(\tau)] \\ C_{2,x}(\tau_1, \tau_2) &= E[x(\tau_1)x(\tau_2)] - E[x(\tau_1)]E[x(\tau_2)] \\ C_{3,x}(\tau_1, \tau_2, \tau_3) &= E[x(\tau_1)x(\tau_2)x(\tau_3)] \\ &- C_{2,x}(\tau_1, \tau_2) - C_{2,x}(\tau_2, \tau_3) - C_{2,x}(\tau_3, \tau_1) - 3C_{1,x}^3(\tau) \end{aligned} \tag{3.22}$$

where $C_{k,x}$ represents the kth-order cumulant of the random process $\boldsymbol{x}$. The first-order and the second-order cumulants are nothing but mean and variance of $\boldsymbol{x}$, respectively. Also, it is shown in Equation 3.22 that the third-order cumulant can be expressed with the third-order moment and the lower-order cumulants. In fact, it is shown in [28] that a kth-order cumulant can be expressed with a kth-order moment as well as $(k-1)$th and lower order cumulants. This means that the kth-order cumulant of the zero-mean random process can be expressed with the kth-order moment, mean and variance.

Now, if $\boldsymbol{x}$ is a Gaussian random process, then kth-order cumulants of $\boldsymbol{x}$ is zero for $k \geq 3$ [28]. In this case, kth-order moments of $\boldsymbol{x}$ can be expressed

as follows.

$$
\begin{aligned}
\mu_k =& 0 \quad (\text{if } k \text{ is an odd number}) \\
\mu_2 =& E[x(\tau_1)x(\tau_2)] \\
\mu_4 =& E[x(\tau_1)x(\tau_2)x(\tau_3)x(\tau_4)] \\
=& \sum_{\substack{i,j,k,l=1: \\ i\neq j\neq k\neq l}}^{4} E[x(\tau_i)x(\tau_j)]E[x(\tau_k)x(\tau_l)] \\
\mu_6 =& E[x(\tau_1)x(\tau_2)x(\tau_3)x(\tau_4)x(\tau_5)x(\tau_6)] \\
=& \sum_{\substack{i,\cdots,n=1: \\ i\neq\cdots\neq n}}^{6} E[x(\tau_i)x(\tau_j)]E[x(\tau_k)x(\tau_l)]E[x(\tau_m)x(\tau_n)]
\end{aligned}
\tag{3.23}
$$

The results listed in Equation 3.22 have used frequently in developing the algorithm explained in the following section.

3.4.2 Proposed Method

In the proposed method, a DUT is first excited once with a pseudo-random sequence ($x(t)$). Then, the Volterra kernels of the DUT are identified by post-processing the DUT responses ($y(t)$) and $x(t)$, and are used to estimate the DUT performance. In this section, a low-complexity algorithm for identifying the Volterra kernels is presented which is aimed at saving the post-processing time. Additionally, a method to maintain good test accuracy regardless of the statistics of the test input is presented. This enables us to use a simple LFSR to generate the pseudorandom sequence.

3.4.2.1 Even-order Volterra Kernel Identification

In order to reduce the complexity, the presented algorithm uses first-order cross-correlations to identify the Volterra kernels instead of using time-consuming higher-order cross-correlations used in conventional methods. This is possible by using various combinations of input sequence products (which called *pattern combinations* in this dissertation) to find the cross-correlation values between the pseudorandom test input and the resulting DUT output.

To identify the even-order Volterra kernels (i.e., the second-order kernel), the *pattern combination* can be used as follows.

$$x^e_{pc}(t) = x(t)x(t+n) \tag{3.24}$$

If we find the cross-correlation of $y(t)$ and $x^e_{pc}(t)$, the responses of all the odd-order Volterra kernels are suppressed due to the property of higher-order moments explained in Equation 3.23. Then, the cross-correlation can be expressed as follows.

$$R^e_{yx}(m) = E[y_2(t)x^e_{pc}(t-m)] = \sum_{\tau_1}\sum_{\tau_2} h_2(\tau_1,\tau_2)\mu^e_4 \tag{3.25}$$

where $y_2(t)$ is the response of the second-order Volterra kernel.

$$y_2(t) = \sum_{\tau_1}\sum_{\tau_2} h_2(\tau_1,\tau_2)x(t-\tau_1)x(t-\tau_2) \tag{3.26}$$

Also, μ^e_4 is the fourth-order moment of x as follows.

$$\mu^e_4 = E[x(t-\tau_1)x(t-\tau_2)x(t-m)x(t+n-m)] \tag{3.27}$$

The value of μ_4^e is determined by n as follows.

$$\mu_4^e = \begin{cases} \sigma_x^4\{\delta(\tau_1 - m)\delta(\tau_2 - m + n) + \delta(\tau_1 - m + n)\delta(\tau_2 - m)\} \\ (\text{if } n \neq 0) \\ \sigma_x^4\{2\delta(\tau_1 - m)\delta(\tau_2 - m) + \delta(\tau_1 - \tau_2)\} \\ (\text{if } n = 0) \end{cases} \tag{3.28}$$

Using the fact that the second-order Volterra kernel is symmetric, and substituting Equation 3.28 into Equation 3.26 yields,

$$\begin{aligned} R_{yx}^{e(i)}(m;n) =& 2\sigma_x^4 h_2(m, m-n) \qquad (\text{if } n \neq 0) \\ R_{yx}^{e(ii)}(m) =& 2\sigma_x^4 h_2(m,m) + \sigma_x^2 E[y_2(t)] \ (\text{if } n = 0) \end{aligned} \tag{3.29}$$

Now, using Equation 3.29, we can find the values of the second-order Volterra kernel as follows.

$$\tilde{h}_2(m, m-n) = \begin{cases} \dfrac{R_{yx}^{e(i)}(m;n)}{2\sigma_x^4} \qquad (\text{if } n \neq 0) \\ \dfrac{R_{yx}^{e(ii)}(m) - \sigma_x^2 E[y_2(t)]}{2\sigma_x^4} \ (\text{if } n = 0) \end{cases} \tag{3.30}$$

Using this algorithm, the total number of cross-correlation calculations required to identify the second-order Volterra kernel with memory length L is $\binom{L}{1}$.

3.4.2.2 Odd-order Volterra Kernel Identification

The following pattern combination, $x_{pc}^o(t)$, is used to identify odd-order Volterra kernels.

$$x_{pc}^o(t) = x(t)x(t+n)x(t+p) \tag{3.31}$$

The cross-correlation of $y(t)$ and $x_{pc}^o(t)$ contains the information of the odd-order Volterra kernels, while the information of all the even-order Volterra

kernels is suppressed due to the property shown in Equation 3.23. Then, the cross-correlation of $y(t)$ and $x^o_{pc}(t)$ can be expressed as follows.

$$\begin{aligned} &R^o_{yx}(m) = E[y(t)x^o_{pc}(t-m)] = \\ &\sum_{\tau} h_1(\tau)\mu^o_4 + \sum_{\tau_1}\sum_{\tau_2}\sum_{\tau_3} h_3(\tau_1,\tau_2,\tau_3)\mu^o_6 \end{aligned} \tag{3.32}$$

where μ^o_4 and μ^o_6 are the fourth-order and the sixth-order moments of x, respectively, as follows.

$$\begin{aligned} \mu^o_4 &= E[x(t-\tau)x(t-m)x(t+n-m)x(t+p-m)] \\ \mu^o_6 &= E[x(t-\tau_1)x(t-\tau_2)x(t-\tau_3)x^o_{pc}(t-m)] \end{aligned} \tag{3.33}$$

To identify the values of the first-order kernel, h_1, and the third-order kernel, h_3, separately from Equation 3.32, a number of different cross-correlation calculations are required listed as below.

$Case1$: If $m \neq n \neq p \neq m$, μ^o_6 can be expressed as follows.

$$\mu^o_6 = 6\sigma^6_x \delta(\tau_1 - m)\delta(\tau_2 - m + n)\delta(\tau_3 - m + p) \tag{3.34}$$

Here, the third-order Volterra kernel is assumed to be symmetric. Also, in this case, $\mu^o_4 = 0$ due to the following property [28].

$$\begin{aligned} E[x(t-\tau)x(t-m)x(t-m+n)x(t-m+p)] = 0 \\ \text{(if } n \neq p,\ n \neq 0 \text{ and } p \neq 0) \end{aligned} \tag{3.35}$$

Then, Equation 3.32 can be expressed as follows.

$$R^{o(i)}_{yx}(m;n,p) = 6\sigma^6_x h_3(m, m-n, m-p) \tag{3.36}$$

$Case2$: If $n = p \neq m$, μ_4^o and μ_6^o can be expressed as follows.

$$\begin{aligned}\mu_4^o =&\sigma_x^4\delta(\tau - m)\\ \mu_6^o =&\sigma_x^6\{9\delta(\tau_1 - m)\delta(\tau_2 - m + p)\delta(\tau_3 - m + p)\\ &+3\sum_{l:l\neq m-p}\delta(\tau_1 - m)\delta(\tau_2 - l)\delta(\tau_3 - l)\}\end{aligned} \tag{3.37}$$

In this case, Equation 3.32 can be expressed as follows.

$$\begin{aligned}R_{yx}^{o(ii)}(m;p) =&\sigma_x^4 h_1(m) + 9\sigma_x^6 h_3(m, m-p, m-p)\\ &+3\sigma_x^6\sum_{l:l\neq m-p} h_3(m, l, l)\end{aligned} \tag{3.38}$$

$Case3$: If $n = p = 0$, μ_4^o and μ_6^o can be expressed as follows.

$$\begin{aligned}\mu_4^o =&3\sigma_x^4\delta(\tau - m)\\ \mu_6^o =&\sigma_x^6\{15\delta(\tau_1 - m)\delta(\tau_2 - m)\delta(\tau_3 - m)\\ &+9\sum_{l:l\neq m}\delta(\tau_1 - m)\delta(\tau_2 - l)\delta(\tau_3 - l)\}\end{aligned} \tag{3.39}$$

and Equation 3.32 can be expressed as follows.

$$R_{yx}^{o(iii)}(m) = 3\sigma_x^4 h_1(m) + 15\sigma_x^6 h_3(m, m, m) + 9\sigma_x^6\sum_{l:l\neq m} h_3(m, l, l) \tag{3.40}$$

$Case4$: Finally, to complete the identification procedure, one more calculation of cross-correlation between the output, $y(t)$, and the input, $x(t)$, is required.

$$\begin{aligned}R_{yx}^{o(iv)}(m) =&E[y(t)x(t-m)]\\ =&\sum_{\tau} h_1(\tau)\mu_2^o + \sum_{\tau_1}\sum_{\tau_2}\sum_{\tau_3} h_3(\tau_1, \tau_2, \tau_3)\mu_4^o\end{aligned} \tag{3.41}$$

In this case, μ_2^o and μ_4^o shown in Equation 3.41 can be expressed as follows.

$$\begin{aligned}\mu_2^o =&\sigma_x^2\delta(\tau - m)\\ \mu_4^o =&\sigma_x^4\{3\delta(\tau_1 - m)\delta(\tau_2 - m)\delta(\tau_3 - m)\\ &+3\sum_{l:l\neq m}\delta(\tau_1 - m)\delta(\tau_2 - l)\delta(\tau_3 - l)\}\end{aligned} \tag{3.42}$$

Thus, in this case, Equation 3.32 can be simplified as follows.

$$R_{yx}^{o(iv)}(m) = \sigma_x^2 h_1(m) + 3\sigma_x^4 h_3(m,m,m) + 3\sigma_x^4 \sum_{l:l\neq m} h_3(m,l,l) \tag{3.43}$$

Now, using Equations 3.36, 3.38, 3.40 and 3.43, the values of the third-order Volterra kernel can be calculated as follows.

$$\tilde{h}_3(m, m-n, m-p) = \begin{cases} \dfrac{1}{6\sigma_x^6}\{R_{yx}^{o(iii)}(m) - 3\sigma_x^2 R_{yx}^{o(iv)}(m)\} \\ (\text{if } n=0, p=0) \\ \dfrac{1}{6\sigma_x^6}\{R_{yx}^{o(i)}(m;n,p)\} \\ (\text{if } n\neq 0, p\neq 0, n\neq p) \\ \dfrac{1}{6\sigma_x^6}\{R_{yx}^{o(ii)}(m;p) - \sigma_x^2 R_{yx}^{o(iv)}(m)\} \\ (\text{if } n=0, p\neq 0) \end{cases} \tag{3.44}$$

Also, the first-order Volterra kernel can be calculated using Equations 3.43 and 3.44 as follows.

$$\begin{aligned} \tilde{h}_1(m) = & \frac{1}{\sigma_x^2}\{R_{yx}^{o(iv)}(m) - 3\sigma_x^4 \tilde{h}_3(m,m,m) \\ & -\sigma_x^4 \sum_{p:p\neq 0} \tilde{h}_3(m, m-p, m-p)\} \end{aligned} \tag{3.45}$$

Using the method presented above, the total number of cross-correlation calculation required to identify the first and the third-order Volterra kernel with memory length L is $\binom{L+2-1}{2} + 1$.

3.4.2.3 Non-Gaussian Pseudorandom Sequence

So far, it is assumed that a pseudorandom sequence has a Gaussian distribution. The accuracy of the identification algorithm is decreased if the

distance of the statistics of pseudorandom sequence from the Gaussian is increased. However, it is difficult to generate a Gaussian-distributed pseudorandom sequence using a general-purpose LFSR, since it requires complex processing to get a true Gaussian distribution, which increases the complexity of the LFSR by a large amount.

If the pseudorandom sequence, $x(t)$, does not have a Gaussian distribution, then the kth-order cumulant of $x(t)$ is not zero for $k \geq 3$ and it can be expressed as follows.

$$C_{k,x}(\tau_1 \cdots \tau_k) = E[x(\tau_1) \cdots x(\tau_k)] - E[g(\tau_1) \cdots g(\tau_k)] \tag{3.46}$$

where $g(t)$ is a Gaussian random process with the same statistics as $x(t)$ [28]. In this case, the higher-order moment equations shown in Equation 3.23, which form base of the presented algorithm, are untenable, and thus, those equations should be modified properly. In order to take the non-Gaussian distribution of the pseudorandom sequence into consideration, the adjust parameter, α_k, is used which measures the difference between the distribution of the pseudorandom sequence and the Gaussian distribution.

$$\alpha_k = \frac{E[x(\tau_1) \cdots x(\tau_k)] - E[g(\tau_1) \cdots g(\tau_k)]}{\sigma_x^k} \tag{3.47}$$

Since the statistics of $x(t)$ and $g(t)$ are known values, we can calculate the values of α_k *a priori* and use it to adjust the values of higher-order moments as follows.

$$\mu_4' = \mu_4 - \alpha_4 \sigma_x^4, \quad \mu_6' = \mu_6 - \alpha_6 \sigma_x^6 \tag{3.48}$$

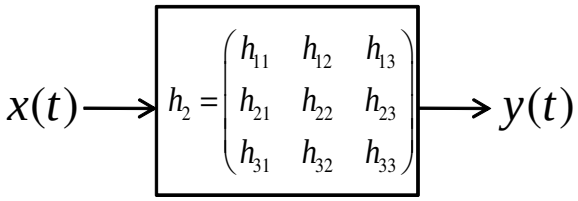


Figure 3.7: Second-order System with Memory Length $L = 3$

Then, by replacing the values of the higher-order moments (μ_4 and μ_6) used in the algorithm with the adjusted values (μ'_4 and μ'_6), we can obtain an accurate result of the Volterra kernel identification even when the pseudorandom sequence does not have an ideal Gaussian distribution.

3.4.2.4 Calculation of Compressed Cross-correlation Series

Using the Volterra kernel identification method presented so far, the total number of cross-correlation calculations required is $\binom{L}{1} + \binom{L+2-1}{2} + 1$. While this number is smaller than the number of calculations required for the conventional method, it still increases quadratically with memory length L which results in increasing the test time. This section presents an algorithm to reduce the time required to identify the Volterra kernels by calculating multiple cross-correlation equations simultaneously.

To explain this method, at first, the brief review of properties of the cross-correlation is presented with an example shown in Figure 3.7. The system, h_2, shown in Figure 3.7 is represented by the second-order transfer function with memory length $L = 3$ while the input to the system, $x(t)$, is a

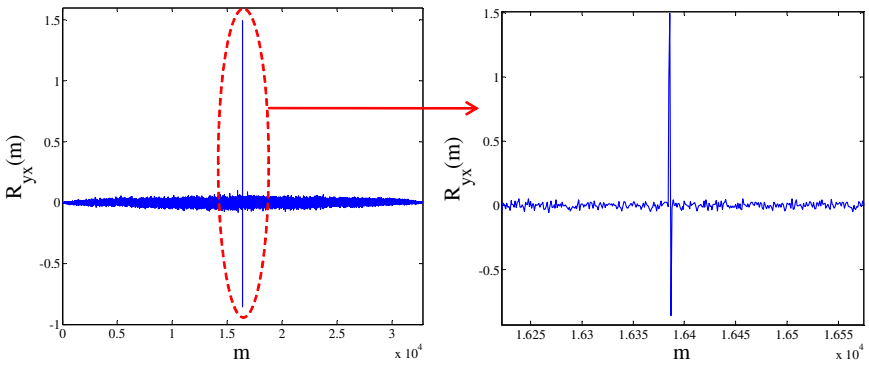
zero-mean maximum-length sequence with length $2^{14}-1$ generated from a 14-bit LFSR. Since $x(t)$ is a maximum-length sequence, the auto-correlation of $x(t)$ ($R_{xx}(m)$) is a close approximation of the Kronecker delta function which exhibits peak value only when $m = 0$. Moreover, the cross-correlation between $x(t)$ and $y(t)$ of the example shown in Figure 3.7 exhibits peak values only for a few points which correspond to the system transfer function, h_2, while exhibits values close to zero (which called *noisy values* in this section) for all other points. This can be seen in Figure 3.8(a) which shows the cross-correlation between $x(t)$ and $y(t)$. In the proposed algorithm, the noisy values shown in Figure 3.8(a) are not required, and thus, two or more series of cross-correlation can be superposed on each other if we can introduce a delay on each cross-correlation series such that the peak values of one series are superposed on the noisy values of other series.

In order to compress multiple cross-correlation series, multiple *pattern combinations*, each with different delay (d_k or d_l) are combined as follows.

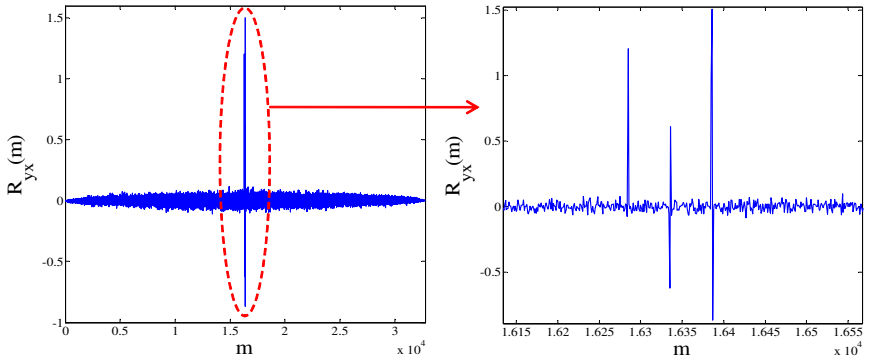
$$x_{cc}(t) = \sum_{l} x_{cd}^{o(l)}(t - d_l) + \sum_{k} x_{cd}^{e(k)}(t - d_k) \tag{3.49}$$

The cross-correlation of $x_{cc}(t)$ and $y(t)$ can be expressed as follows.

$$\begin{aligned}
&E[y(t)x_{cc}(t-m)] = \sum_{k}\sum_{\tau_1}\sum_{\tau_2} h_2(\tau_1,\tau_2)\mu_4^{(k)} \\
&+ \sum_{l}\{\sum_{\tau} h_1(\tau)\mu_4^{(l)} + \sum_{\tau_1}\sum_{\tau_2}\sum_{\tau_3} h_3(\tau_1,\tau_2,\tau_3)\mu_6^{(l)}\} \\
&\mu_4^{(k)} = E[x(t-\tau_1)x(t-\tau_2)x_{pc}^{e(k)}(t-m-d_k)] \\
&\mu_4^{(l)} = E[x(t-\tau)x_{pc}^{o(l)}(t-m-d_l)] \\
&\mu_6^{(l)} = E[x(t-\tau_1)x(t-\tau_2)x(t-\tau_3)x_{pc}^{o(l)}(t-m-d_l)]
\end{aligned} \tag{3.50}$$



(a) Cross-correlation of $x(t)$ and $y(t)$



(b) Cross-correlation of $x_{cc}(t)$ and $y(t)$

Figure 3.8: Cross-correlation Example

Each term of the summation in Equation 3.50 represents an independent cross-correlation equation (such as Equation 3.25 or 3.32) delayed by d_k or d_l. Thus, Equation 3.50 contains $k + l$ different cross-correlation equations which are calculated together under one combined equation.

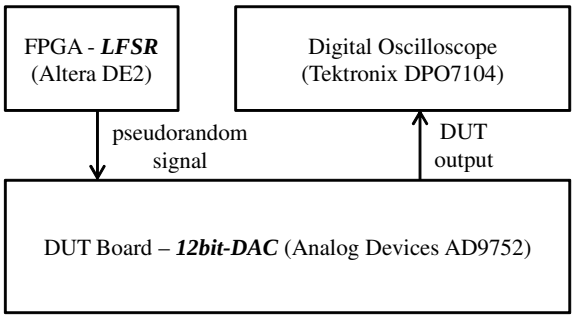
Let us apply the compressed cross-correlation method to the example shown in Figure 3.7. Based on the algorithm presented in the Sections 3.4.2.2 and 3.4.2.3, it is required to calculate three different cross-correlation equations to identify the second-order transfer function with memory length 3 featured in this example. Then, we can use the following sequence to find the values of three different cross-correlation series simultaneously.

$$x_{cc}(t) = x(t)^2 + x(t)x(t - 1 - 50) + x(t)x(t - 2 - 50 * 2) \tag{3.51}$$

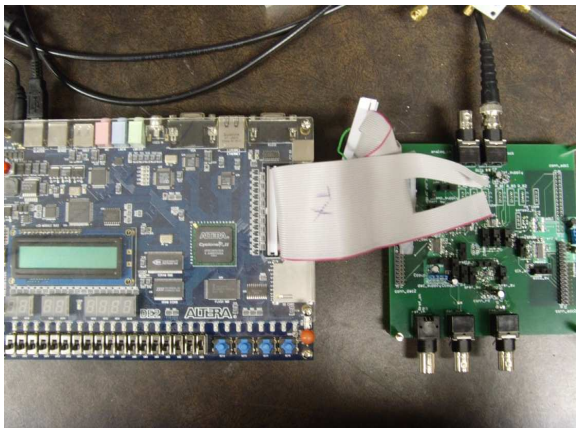
Figure 3.8(b) shows the cross-correlation between $x_{cc}(t)$ and $y(t)$. As we can see from that figure, by introducing the delay of 50 between each cross-correlation series, we can combine multiple cross-correlation series into one equation, and can acquire the required information out of the combined equation.

3.4.3 Experimental Results

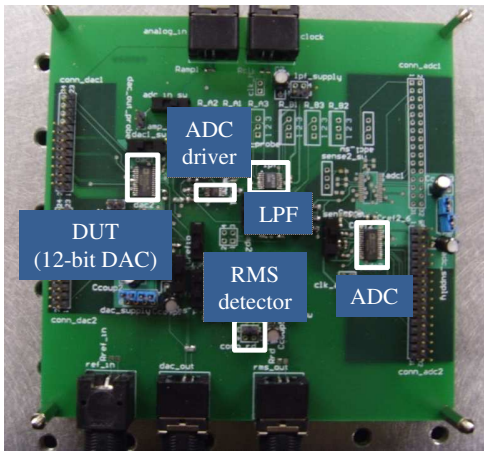
The method described in this section was applied to characterize the performance of a 14-bit DAC from Analog Devices™. Figure 3.9 depicts the measurement setup. To generate a pseudorandom sequence, a 12-bit LFSR was designed and loaded on to a FPGA board. The LFSR was built using a



(a) Hardware Measurement Setup



(b) FPGA board and DUT board



(c) Configuration of the DUT board

Figure 3.9: Measurement Setup

14-stage maximum-length sequence generator and phase shifters as shown in Figure 3.10. The phase shifters were used to break the correlation between the subsequent patterns generated by the maximum-length sequence generator. The LFSR used in this measurement required only a few tens of logic gates and latches. The total area is $465\mu m^2$ in case a 130nm technology is used. Thus, it can be observed the area overhead of the LFSR is small for practical cases. The analog output of the DAC was captured by a digital oscilloscope and post-processed using MATLAB.

Figure 3.11 shows the test and validation flow. First, specifications of the DUT (SNR and THD) were predicted using the estimated DUT model built from the pseudorandom test method. Next, to validate the predicted specifications, actual specifications of the DUT were measured using the sinusoidal wave (test stimulus) generated from the FPGA board. Noise power of the DUT is required in calculating the SNR. We can calculate the noise power by analyzing the DUT output in response to a pair of identical pseudorandom sequence (that are applied the DUT subsequently) as described in Section 3.3.2.

Table 3.4 (25MHz sampling) and 3.5 (50MHz sampling) summarize the predicted and the actual values of SNR and THD for various cases with different input amplitudes and frequencies. In the experiment shown in each table, the test method developed in this research required a single measurement with the pseudorandom sequence to predict the specification values for various cases. On the other hand, the conventional test method requires sepa-

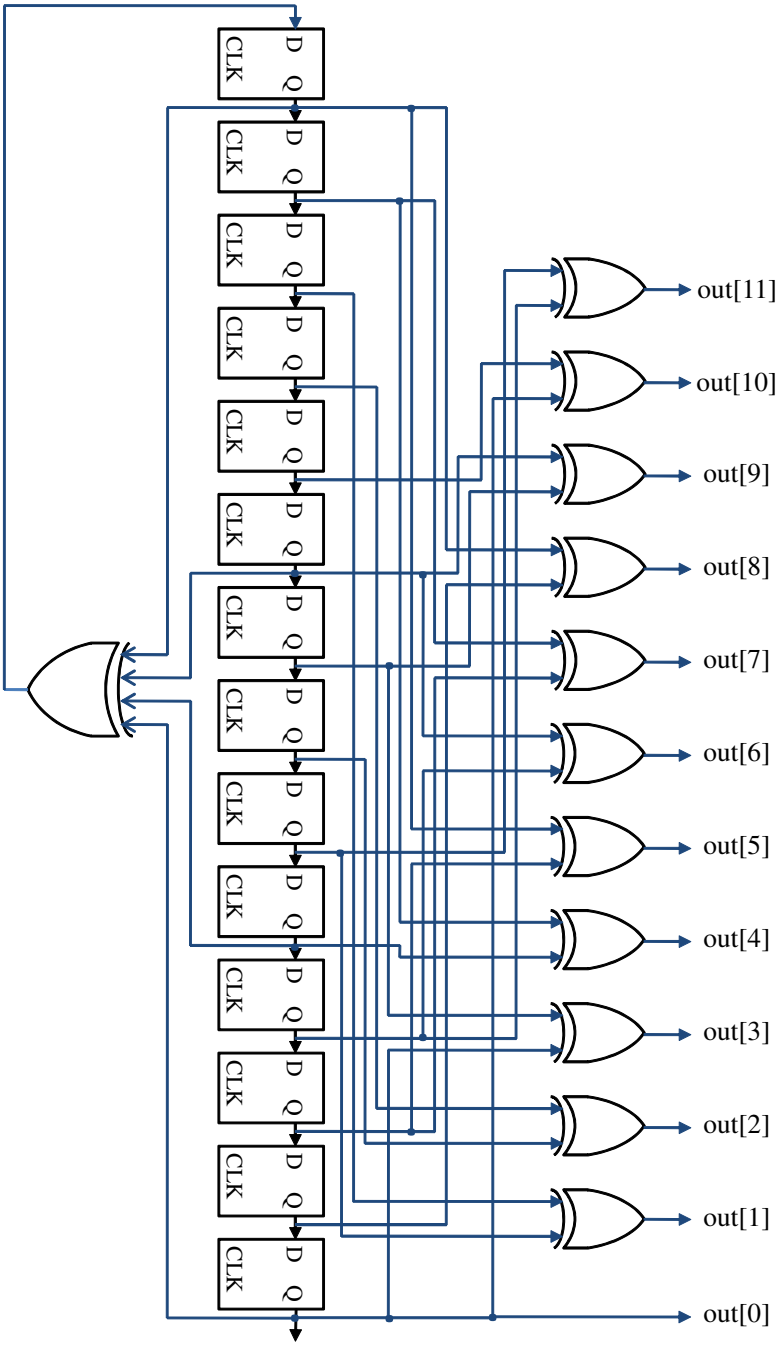


Figure 3.10: 12-bit Pseudorandom Pattern Generator

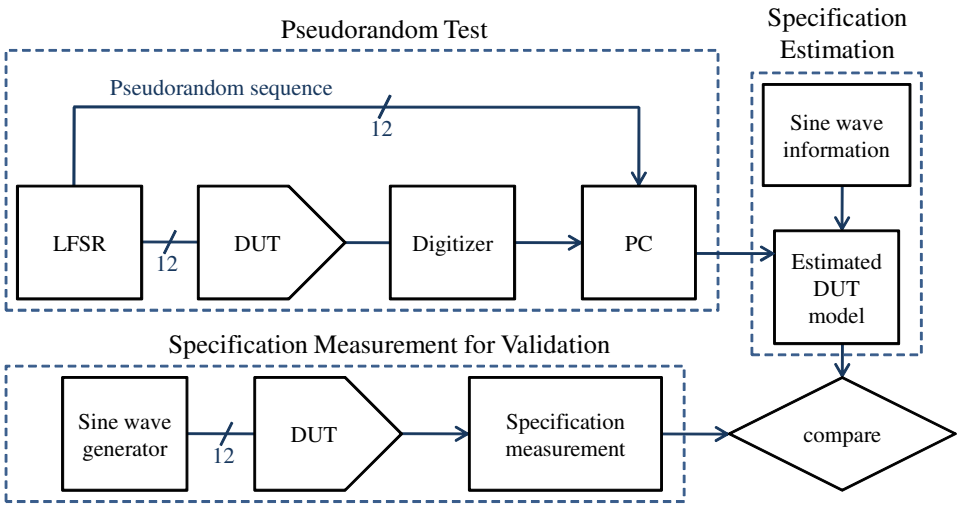


Figure 3.11: Test and Validation Flow

rate measurements for each case, which results in multiple measurements with different sinusoidal inputs. As we can see from the tables, the predicted and the actual values are well correlated for most cases. All the prediction errors were held within 3.6dB of the actual values for the DUT[1].

In the experiment described above, a Volterra series model with a memory length of 5 is used, which meant that we needed to calculate a total of 21 cross-correlation equations. This number is reduced by using the compressed cross-correlation method and calculating multiple cross-correlation values simultaneously. Figure 3.12 depicts the measurement results with different numbers of cross-correlation calculations. A lower number of cross-correlation cal-

[1]Note that the specification values shown in the Table 3.4 and 3.5 are smaller than those in the actual specification sheet; this is because the signal generated from the FPGA board has an effective resolution of 8 bits.

Table 3.4: Hardware Measurement Data - the DUT operating at 25MHz

Input Frequency	Input Amplitude	Specification	Actual value (dB)	Predicted value (dB)	Error (dB)
50kHz	0dBFS	SNR	34.77	32.44	2.33
		THD	−27.65	−26.69	0.96
50kHz	−6dBFS	SNR	28.79	26.05	2.73
		THD	−28.01	−28.77	0.76
250kHz	0dBFS	SNR	33.76	31.82	1.95
		THD	−37.81	−37.24	0.57
250kHz	−6dBFS	SNR	26.93	25.07	1.86
		THD	−38.31	−41.89	3.58
500kHz	0dBFS	SNR	33.43	31.02	2.41
		THD	−36.17	−37.36	1.19
500kHz	−6dBFS	SNR	27.54	25.00	2.54
		THD	−45.01	−43.07	1.94

Table 3.5: Hardware Measurement Data - the DUT operating at 50MHz

Input Frequency	Input Amplitude	Specification	Actual value (dB)	Predicted value (dB)	Error (dB)
100kHz	0dBFS	SNR	34.49	32.28	2.21
		THD	−26.22	−24.31	1.91
100kHz	−6dBFS	SNR	28.61	26.03	2.58
		THD	−21.32	−21.14	0.18
250kHz	0dBFS	SNR	33.91	31.46	2.45
		THD	−36.56	−35.77	0.79
250kHz	−6dBFS	SNR	28.06	25.39	2.67
		THD	−35.99	−35.18	0.81
500kHz	0dBFS	SNR	33.51	31.28	2.23
		THD	−38.44	−37.09	1.35
500kHz	−6dBFS	SNR	27.60	25.12	2.48
		THD	−34.34	−37.67	3.33

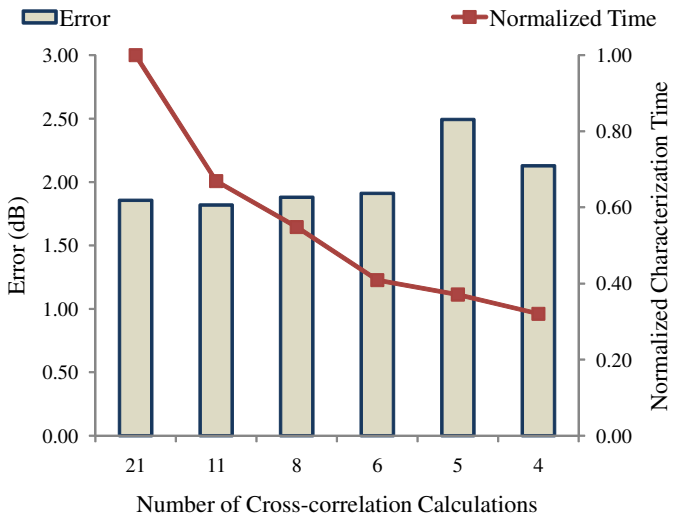


Figure 3.12: Characterization Time Reduction and Prediction Error

culations means more cross-correlation series were compressed together and calculated simultaneously. The bar graph exhibits the average prediction errors of the THD estimation for the various cases shown in Table 3.4, and the line graph exhibits the processing time which is normalized to that of the case when 21 cross-correlation calculations were required (i.e., no compression was made). From Figure 3.12, it can be observed that the characterization time decreased by 70% using the new method while the error increased only by a small amount.

3.4.4 Summary

In Section 3.4, an efficient pseudorandom test method for nonlinear AMS circuits has been discussed. The method described in this section uses

a Volterra series to model nonlinear behaviors of a DUT accurately. A pseudorandom signal is used as a test stimulus to excite the nonlinear DUT and find parameters of the Volterra series which are used to estimate the various performance parameters of the DUT. Since the presented method uses a single excitation of the DUT with a pseudorandom signal to characterize its performance at multiple frequencies, the test time is reduced considerably. In addition, the pseudorandom signal is easily generated using a LFSR which makes the presented pseudorandom test method extendable to embedded AMS circuits without increasing the test cost. Experimental results show that the method developed in this research estimates the performance of a DUT within 3.6dB of the actual values.

Chapter 4

Parallel Test of Analog and Mixed-Signal Circuits

This chapter presents parallel test algorithms for AMS circuits and discusses how these algorithms can be used to save the test time in finding the performance parameters of the multiple DUTs. The algorithms presented in this chapter aim to increase the test throughput by testing multiple DUTs at the same time using a common test setup while not compromising the test accuracy.

There are two parallel test methods presented in this chapter. The first method (*parallel loopback test*) presented in Section 4.1 configures multiple loopback paths using pairs of ADCs and DACs. In this method, a single-tone sinusoidal signal is used as a test stimulus and simple analog circuits are used to solve the fault masking problem. The second method (*parallel pseudorandom test*) presented in Section 4.2 employs a pseudorandom signal to test multiple AMS circuits *simultaneously*, and uses statistical properties of the pseudorandom signal to solve the fault masking issue. The parallel pseudorandom test requires only a single measurement to test multiple DUTs, and thus the test throughput can be increased by a factor of N compared

to conventional test method, where N is the total number of DUTs tested simultaneously. In contrast, in the parallel loopback test, the test throughput can be increased up to a factor of two. Hence, the parallel loopback test can be used in cases where the digital-in/digital-out test environment is desired to reduce the test cost along with increasing the test throughput while the parallel pseudorandom test can be used in cases where increasing the test throughput is a primary objective.

4.1 Parallel Loopback Test of Mixed-Signal Circuits

This section presents an efficient parallel test algorithm, called *parallel loopback test* which increases the level of parallelism in a mixed-signal test without suffering from the fault masking problem. This test algorithm is targeted to parallel testing of multiple mixed-signal circuits in a loopback mode. The example of testing a DAC and an ADC in the loopback mode is presented in the following section. In the parallel loopback test, multiple sets of the ADC/DAC pairs are tested in parallel using a common DUT board which consists of a simple analog adder and an RMS detector. Outputs of the DACs are connected to the inputs of the ADCs through the DUT board to form the loopback path. The resulting test setup has digital inputs and digital outputs, so expensive analog waveform generators and digitizers are not required in this test algorithm. The test input comes from a digital signal generator which can be shared among multiple DUTs. The individual performance parameters of the DUTs are calculated by measuring and post-processing the loopback

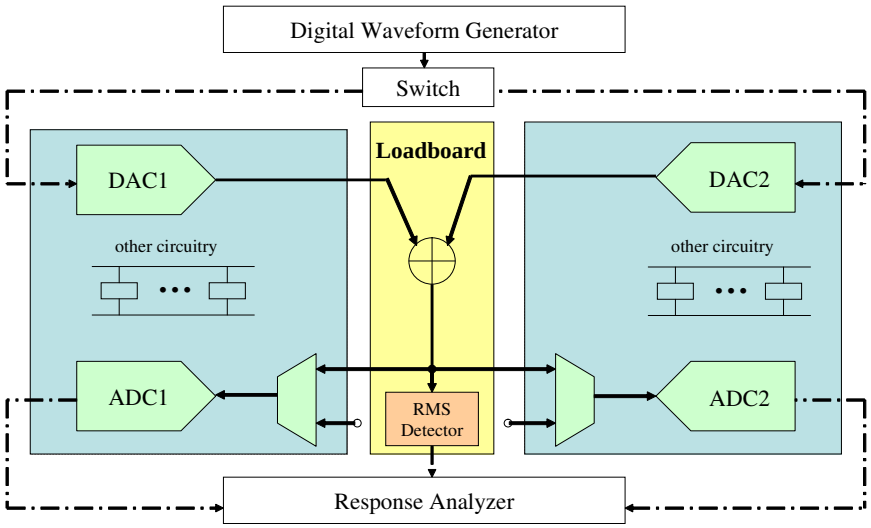


Figure 4.1: Proposed Parallel Loopback Test Scheme

responses captured at the output of the ADCs. The analog adder and the RMS detector on the DUT board are used to extract the performance parameters of multiple DUTs separately and to suppress the effect of the fault masking. This technique can help reduce the cost and time of testing mixed-signal circuits without compromising the test accuracy.

4.1.1 Harmonic Distortion Calculation

This section presents the parallel loopback test algorithm to characterize the harmonic distortion of DUTs. The explanations of the algorithm first starts with two sets of ADC/DAC pairs and later extend the algorithm to test more than two sets of ADC/DAC pairs. Figure 4.1 shows the parallel loopback test setup where two sets of ADCs and DACs are externally connected

to the DUT board. In manufacturing test, we can use a DUT board or probe card as the DUT board as shown in Figure 4.1. The DUT board has a simple analog adder and an RMS detector which can be characterized prior to use. Outputs of both DACs are connected to the adder and the output of the adder is routed to both ADCs. Thus, the DUTs share a common loopback path and there is one module, the analog adder, placed on the loopback path. Since the analog adder can be easily designed to have good linearity [44], we can assume that the harmonic distortion introduced by the adder is negligible[2]. The input to each DAC is illustrated in Figure 4.2. The t notation is used in the digital domain for simplicity. It can be seen that the input to both DACs are the same except for the interval where there is no input to the DAC2 (time interval $\mathcal{A}$) and the DAC1 (time interval $\mathcal{B}$). Thus, we can use one waveform generator and a switch to route the input signal to each DAC.

To find the harmonic distortion parameters, first a sine wave input, $A\cos(\omega t)$, with amplitude A is applied to the DAC1 while the input signal path to the DAC2 is disconnected at the switch (time interval $\mathcal{A}$). The amplitude can have any value as long as it does not saturate the ADCs during the time interval $\mathcal{C}$. Figure 4.2(a) shows the test setup during the time interval $\mathcal{A}$. The outputs of the loopback path I and II are as follows.

$$\hat{y}_{lb1}(t) = y_{lb1}(t) + n_{lb1}(t),\ \hat{y}_{lb2}(t) = y_{lb2}(t) + n_{lb2}(t) \tag{4.1}$$

[2]Even if the adder introduces some nonlinearity, this can be readily characterized since the adder is implemented on the DUT board.

where $n_{lb1}(t)$ and $n_{lb2}(t)$ are the output noise of the loopback paths I and II respectively. $y_{lb1}(t)$ and $y_{lb2}(t)$ are Taylor series expansions which can be expressed as follows.

$$\begin{aligned} y_{lb1}(t) &= \delta_{11} A\cos(\omega t) + \delta_{12} A^2 \cos^2(\omega t) + \delta_{13} A^3 \cos^3(\omega t) \\ &= (\delta_{11} A + \frac{3\delta_{13}}{4} A^3)\cos(\omega t) + \frac{\delta_{12}}{2} A^2 \cos(2\omega t) + \frac{\delta_{13}}{4} A^3 \cos(3\omega t) \end{aligned} \tag{4.2}$$

$$y_{lb2}(t) = \delta_{21} A\cos(\omega t) + \delta_{22} A^2 \cos^2(\omega t) + \delta_{23} A^3 \cos^3(\omega t) \tag{4.3}$$

where the constants δ_{11} - δ_{23} are as follows.

$$\begin{aligned} &\delta_{11} = \alpha_1\gamma_1, \quad \delta_{12} = \gamma_1\alpha_2 + \gamma_2\alpha_1^2, \\ &\delta_{13} = \gamma_1\alpha_3 + 2\gamma_2\alpha_1\alpha_2 + \gamma_3\alpha_1^3, \\ &\delta_{21} = \alpha_1\theta_1, \quad \delta_{22} = \theta_1\alpha_2 + \theta_2\alpha_1^2, \\ &\delta_{23} = \theta_1\alpha_3 + 2\theta_2\alpha_1\alpha_2 + \theta_3\alpha_1^3 \end{aligned} \tag{4.4}$$

In the above equations, α_i, γ_i and θ_i are the ith harmonic distortion coefficients of the DAC1, ADC1 and ADC2 respectively[3]. Since we already know the value of the input amplitude, A, we can find the values of δ_{11} - δ_{23} by measuring the frequency response at ω, 2ω and 3ω. Next, the sine wave input is applied to the DAC2 while the input signal path to the DAC1 is disconnected (time interval $\mathcal{B}$). The Taylor series expansion of the output of the loopback path III, shown in Figure 4.2(b), is as follows.

$$y_{lb3}(t) = \delta_{31} A\cos(\omega t) + \delta_{32} A^2 \cos^2(\omega t) + \delta_{33} A^3 \cos^3(\omega t) \tag{4.5}$$

[3]In this section, harmonic distortions up to the third order are considered. However, the presented performance characterization method itself is not limited to the third order and it is straightforward to extend to higher orders.

where the constants δ_{31}, δ_{32} and δ_{33} are as follows.

$$\begin{aligned} \delta_{31} &= \beta_1\gamma_1, \quad \delta_{32} = \gamma_1\beta_2 + \gamma_2\beta_1^2, \\ \delta_{33} &= \gamma_1\beta_3 + 2\gamma_2\beta_1\beta_2 + \gamma_3\beta_1^3, \end{aligned} \tag{4.6}$$

and β_i are the ith harmonic distortion coefficients of the DAC2. Finally, the sine wave input is applied to both DACs during time interval $\mathcal{C}$. Then, the Taylor series expansion of the output of the loopback path IV is as follows.

$$\begin{aligned} y_{lb4}(t) \;=\; & \delta_{41}A\cos(\omega t) + (\delta_{12} + \delta_{32} + 2\delta_{42})A^2\cos^2(\omega t) \\ & + (\delta_{13} + \delta_{33} + \delta_{43})A^3\cos^3(\omega t) \end{aligned} \tag{4.7}$$

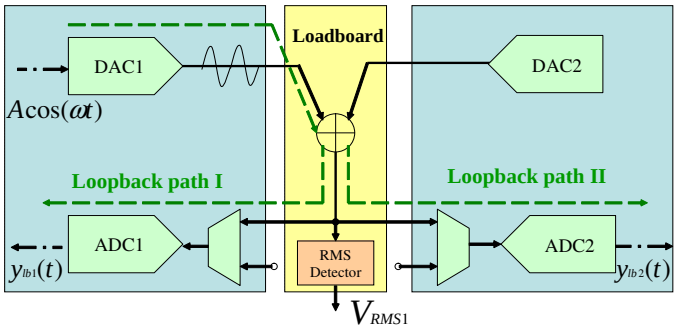
where three constants δ_{41}, δ_{42} and δ_{43} are as follows.

$$\begin{aligned} \delta_{41} &= (\alpha_1 + \beta_1)\gamma_1, \quad \delta_{42} = \gamma_2\alpha_1\beta_1, \\ \delta_{43} &= 3\gamma_3\alpha_1\beta_1(\alpha_1 + \beta_1) + 2\gamma_2(\alpha_1\beta_2 + \beta_1\alpha_2) \end{aligned} \tag{4.8}$$

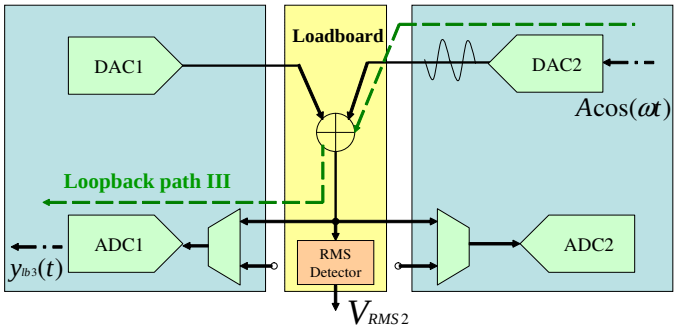
Using Equations 4.4, 4.6 and 4.8, we can formulate 11 linearly independent equations. This means that we do not have sufficient equations to find the values of all 12 harmonic distortion coefficients, α_{1-3}, β_{1-3}, γ_{1-3} and θ_{1-3}. Instead of finding absolute values of all 12 coefficients directly, we can first express 11 coefficients in terms of the 1 remaining coefficient called the *reference variable*. In the remainder of this section, α_1 is used as the reference variable. After several steps of calculations, we can formulate the following equations from Equations 4.4, 4.6 and 4.8.

$$\begin{bmatrix} \alpha_1 & \beta_1 \\ \alpha_2 & \beta_2 \\ \alpha_3 & \beta_3 \end{bmatrix} = \begin{bmatrix} 1 & C_{\beta_1} \\ C_{\alpha_2} & C_{\beta_2} \\ C_{\alpha_3} & C_{\beta_3} \end{bmatrix} \begin{bmatrix} \alpha_1 & 0 \\ 0 & \alpha_1 \end{bmatrix} \tag{4.9}$$

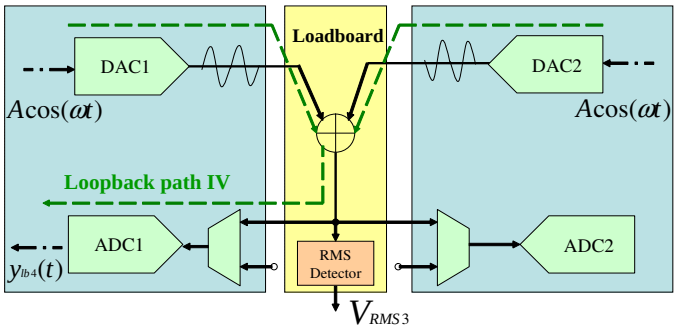
$$\begin{bmatrix} \gamma_1 & \theta_1 \\ \gamma_2 & \theta_2 \\ \gamma_3 & \theta_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{\alpha_1} & 0 & 0 \\ 0 & \frac{1}{\alpha_1^2} & 0 \\ 0 & 0 & \frac{1}{\alpha_1^3} \end{bmatrix} \begin{bmatrix} C_{\gamma_1} & C_{\theta_1} \\ C_{\gamma_2} & C_{\theta_2} \\ C_{\gamma_3} & C_{\theta_3} \end{bmatrix} \tag{4.10}$$



(a) Time Interval $\mathcal{A}$: Test input to DAC1 only



(b) Time Interval $\mathcal{B}$: Test input to DAC2 only



(c) Time Interval $\mathcal{C}$: Test input to DAC1 & DAC2

Figure 4.2: Test Setup for Three Time Intervals

where the values of $C_{\alpha_{2-3}}$, $C_{\beta_{1-3}}$, $C_{\gamma_{1-3}}$ and $C_{\theta_{1-3}}$ can be expressed as Equation 4.11.

$$
\begin{aligned}
&C_{\gamma_1} = \delta_{11}, C_{\beta_1} = \frac{\delta_{31}}{\delta_{11}}.C_{\theta_1} = \delta_{21}, C_{\gamma_2} = \frac{\delta_{11}\delta_{42}}{\delta_{31}}, C_{\alpha_2} = \frac{\delta_{12} - C_{\gamma_2}}{C_{\gamma_1}}, C_{\beta_2} = \frac{\delta_{32} - C_{\beta_1}^2 C_{\gamma_2}}{C_{\gamma_1}}, \\
&C_{\gamma_3} = \frac{\delta_{43} - 2C_{\gamma_2}(C_{\beta_2} + C_{\beta_1}C_{\alpha_2})}{3C_{\beta_1}(1 + C_{\beta_1})}, C_{\alpha_3} = \frac{\delta_{13} - C_{\gamma_3} - 2C_{\gamma_2}C_{\alpha_2}}{C_{\gamma_1}}, \\
&C_{\theta_2} = \delta_{22} - C_{\theta_1}C_{\alpha_2}, C_{\beta_3} = \frac{\delta_{33} - C_{\beta_1}^3 C_{\gamma_3} - 2C_{\gamma_2}C_{\beta_2}C_{\beta_1}}{C_{\gamma_1}}, C_{\theta_3} = \delta_{23} - C_{\theta_1}C_{\alpha_3} - 2C_{\theta_2}C_{\alpha_2}
\end{aligned}
\tag{4.11}
$$

Now, if we can find the value of the *reference variable* (α_1), all the harmonic distortion coefficients can be calculated using Equations 4.9, 4.10 and 4.11. In the parallel loopback test algorithm, an RMS detector is used to find the value of the reference variable, α_1. Using current technology, the RMS detector can be designed to operate at speeds up to a few GHz while the detection error can be held less than 5% [5, 52]. Also, the output of the RMS detector has a DC value, and thus it can be measured using low-cost test equipment.

The DC values measured at the output of the RMS detector during the time intervals $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ can be expressed as follows.

$$
\begin{aligned}
V_{RMS1} \;=\; &[(\frac{1}{2}A^2 + \frac{3}{8}(C_{\alpha 2}^2 + 2C_{\alpha 3})A^4 \\
&+ \frac{5}{16}C_{\alpha 3}^2 A^6)\alpha_1^2 + \overline{v_{DAC1}^2}]^{0.5}
\end{aligned}
\tag{4.12}
$$

$$
\begin{aligned}
V_{RMS2} \;=\; &[(\frac{1}{2}C_{\beta 1}^2 A^2 + \frac{3}{8}(C_{\beta 2}^2 + 2C_{\beta 3})A^4 \\
&+ \frac{5}{16}C_{\beta 3}^2 A^6)\alpha_1^2 + \overline{v_{DAC2}^2}]^{0.5}
\end{aligned}
\tag{4.13}
$$

$$
\begin{aligned}
V_{RMS3} \;=\; &[(\frac{1}{2}(1 + C_{\beta 1})^2 A^2 + \frac{3}{8}(C_{\alpha\beta 2}^2 + 2C_{\alpha\beta 3})A^4 \\
&+ \frac{5}{16}C_{\alpha\beta 3}^2 A^6)\alpha_1^2 + \overline{v_{DAC1}^2} + \overline{v_{DAC2}^2}]^{0.5}
\end{aligned}
\tag{4.14}
$$

Where $C_{\alpha\beta 2} = C_{\alpha 2} + C_{\beta 2}$ and $C_{\alpha\beta 3} = C_{\alpha 3} + C_{\beta 3}$. Also, $\overline{v_{DAC1}^2}$ and $\overline{v_{DAC2}^2}$ are the output noise powers of the DAC1 and DAC2, respectively. It is assumed that the output noise of DAC1 and DAC2 are uncorrelated to each other. Equations 4.12-4.14 are linearly independent, while there are three unknown variables, α_1, $\overline{v_{DAC1}^2}$ and $\overline{v_{DAC2}^2}$, in these equations. So, we can find the value of α_1 using Equations 4.12-4.14, and successively find the remaining values of the harmonic distortion coefficients, α_{2-3}, β_{1-3}, γ_{1-3} and θ_{1-3}, using Equations 4.9, 4.10 and 4.11.

The adder and the RMS detector used in this test algorithm can work with different input frequencies without changing the configuration. This means that this algorithm can be flexibly applied to various test setups without reconfiguring the DUT board, and thus help reduce the test cost.

4.1.2 Noise Power Calculation

This section describes the algorithm to find the noise power of DUTs using the parallel loopback test.

First, assume that $\sqrt{K_{\alpha\gamma}}$ is the overall gain of the loopback path which consists of DAC1 and ADC1 (loopback path I at time interval $\mathcal{A}$). The value of $\sqrt{K_{\alpha\gamma}}$ can be calculated using the harmonic distortion coefficients found using the algorithm explained in Section 4.1.1. Also, assume that $n_\alpha(t)$ and $n_\gamma(t)$ are the output referred noise of the DAC1 and ADC1 respectively. Then, the output of the loopback path I can be expressed as follows.

$$\hat{y}_{lb1}(t) = y_{lb1}(t) + \sqrt{K_{\alpha\gamma}} n_\alpha(t) + n_\gamma(t) \tag{4.15}$$

where $y_{lb1}(t)$ is given in Equation 4.3 and two noise components, $n_\alpha(t)$ and $n_\gamma(t)$, are assumed to be uncorrelated with each other. By performing frequency analysis at the output of the loopback path I, we can extract the noise components from the signal tone and its harmonics, and calculate the noise power [40]. The calculated noise power, $\overline{v_{lb1}^2}$, can be expressed as follows.

$$\overline{v_{lb1}^2} = \int_0^\infty K_{\alpha\gamma} N_\alpha(f) df + \int_0^\infty N_\gamma(f) df \tag{4.16}$$

where $N_\alpha(f)$ and $N_\beta(f)$ are the Power Spectral Density (PSD) of the output referred noise of the DAC1 and ADC1 respectively. Applying similar approaches to the loopback path II at the time interval $\mathcal{A}$, the loopback path III at the time interval $\mathcal{B}$ and the loopback path IV at the time interval $\mathcal{C}$, we can formulate the following equations.

$$\overline{v_{lb2}^2} = \int_0^\infty K_{\alpha\theta} N_\alpha(f) df + \int_0^\infty N_\theta(f) df \tag{4.17}$$

$$\overline{v_{lb3}^2} = \int_0^\infty K_{\beta\gamma} N_\beta(f) df + \int_0^\infty N_\gamma(f) df \tag{4.18}$$

$$\overline{v_{lb4}^2} = \int_0^\infty [K_{\alpha\gamma} N_\alpha(f) + K_{\beta\gamma} N_\beta(f)] df + \int_0^\infty N_\gamma(f) df \tag{4.19}$$

where $N_\beta(f)$ and $N_\theta(f)$ are the noise PSD of the DAC2 and ADC2 respectively, and $\overline{v_{lb2}^2}$, $\overline{v_{lb3}^2}$ and $\overline{v_{lb4}^2}$ are the output noise power of the loopback paths II, III and IV respectively. Also, $K_{\alpha\theta}$ and $K_{\beta\gamma}$ are the overall gain of the loopback paths consisting of DAC1/ADC2 and DAC2/ADC1, respectively. Now, using the Equations 4.16 - 4.19, and given $K_{\alpha\gamma}$, $K_{\alpha\theta}$ and $K_{\beta\gamma}$, the noise power of

each DUT can be calculated as follows.

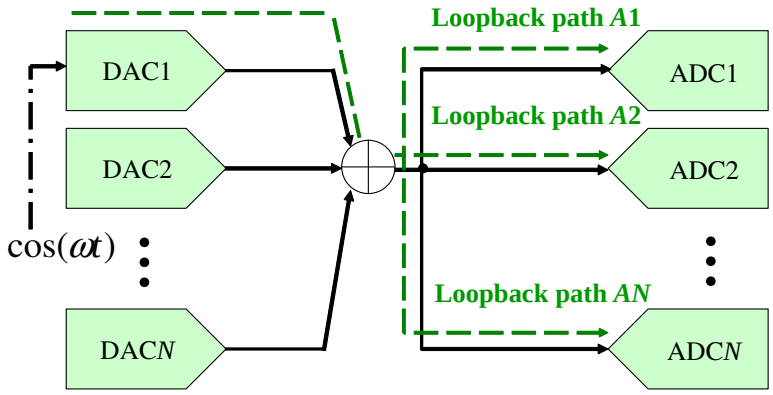
$$\begin{aligned}
\int_0^\infty N_\alpha(f)df &= \frac{\overline{v_{lb4}^2} - \overline{v_{lb3}^2}}{K_{\alpha\gamma}} \\
\int_0^\infty N_\beta(f)df &= \frac{\overline{v_{lb4}^2} - \overline{v_{lb1}^2}}{K_{\beta\gamma}} \\
\int_0^\infty N_\gamma(f)df &= \overline{v_{lb1}^2} + \overline{v_{lb3}^2} - \overline{v_{lb4}^2} \\
\int_0^\infty N_\theta(f)df &= \overline{v_{lb2}^2} - \frac{K_{\alpha\theta}}{K_{\alpha\gamma}}(\overline{v_{lb4}^2} - \overline{v_{lb3}^2})
\end{aligned} \tag{4.20}$$

Now that we have found the harmonic distortion parameters and the noise power, we can calculate the performance parameters of each DUT such as SNR, SNDR, THD, etc. [11]. Also, these results can be used to characterize the amplitude mismatch between different ADCs or different DACs which is an important parameter in RF/Audio CODEC where there are normally two sets of the ADCs and DACs used for the I and Q channels.

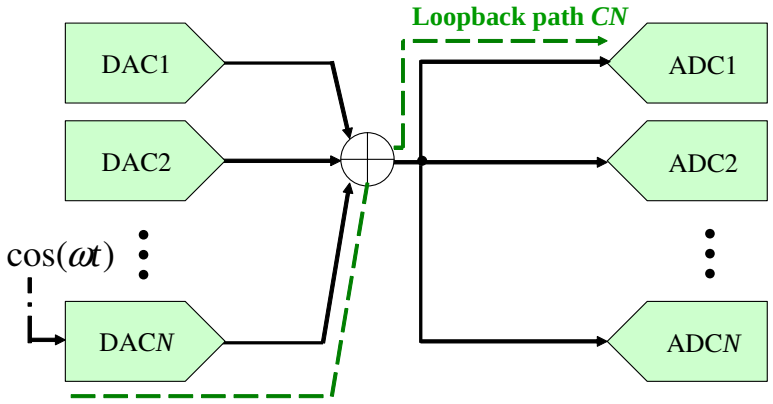
4.1.3 Parallel Loopback Test Algorithm for Multiple DUTs

So far, the parallel loopback test algorithm has been presented which can test two sets of ADC/DAC pairs in parallel using a common test equipment including a DUT board. This algorithm can be easily extended to the case where there are more than two sets of ADC/DAC pairs. This section describes the extension of the parallel loopback test algorithm that can be used to test multiple DUTs with a common test equipment.

Figure 4.3 shows the case where there are N sets of ADC/DAC pairs tested in parallel ($2N$ DUTs total). The maximum number of DUTs that can



(a) Test Setup for ADCs



(b) Test Setup for DACs

Figure 4.3: Application of the Parallel Loopback Test to Multiple DUTs

be tested in parallel depends on the driving capacity of the analog adder the dynamic range of the ADCs. In this section, it is assumed that the adder can drive N DUTs in parallel and the output of the analog adder does not saturate the ADC. Also, note that, although the example of testing the same number of the DACs and ADCs is presented in this section, this number does not have to be same.

First, the harmonic distortion coefficients and the noise power of the DAC1, DAC2, ADC1 and ADC2 are measured using the procedure described in the previous sections. Next, the harmonic distortion coefficients of all the remaining ADCs can be characterized by examining the measured output responses at each ADC during the time interval $\mathcal{A}$. Note that this step does not require additional test inputs to the DUTs. For example, the following equation describes the output of the ADCN at the time interval $\mathcal{A}$.

$$\begin{aligned} y_{lbN}(t) =& \alpha_1\rho_1\cos(\omega t) + (\rho_1\alpha_2 + \rho_2\alpha_1^2)\cos^2(\omega t) \\ &+ (\rho_1\alpha_3 + 2\rho_2\alpha_1\alpha_2 + \rho_3\alpha_1^3)\cos^3(\omega t) \end{aligned} \tag{4.21}$$

where ρ_i is the ith harmonic distortion coefficient of the ADCN. Since we already know the value of the α_1, α_2 and α_3, we can calculate the harmonic distortion coefficients of the ADCN from Equation 4.21. The output noise power of the loopback path AN shown in Figure 4.3(a) can also be calculated and expressed as follows.

$$\overline{v_{lbN}^2} = \int_0^\infty K_{\alpha\rho}N_\alpha(f)df + \int_0^\infty N_\rho(f)df \tag{4.22}$$

Now using the values of $\int_0^\infty N_\alpha(f)df$, $\overline{v_{lbN}^2}$ and $K_{\alpha\rho}$ which we already know, we

can calculate the noise power of the DACN, $\int_0^\infty N_\rho(f)df$. The same approach can be applied to all the remaining ADCs.

We can use a similar approach to find the harmonic distortion coefficients and the noise power of the remaining DACs. The difference is that, this time, a digital sine wave input, $\cos(\omega t)$, should be applied to the DAC that we want to test. For example, to find the performance parameters of the DACN shown in Figure 4.3(b), the sine wave input is applied to the DACN and the output of the loopback path CN, $y_{lbCN}(t)$, is captured at the ADC1. Now, by post-processing the output, $y_{lbCN}(t)$, we can formulate the Taylor series expansion and noise equation similar to Equations 4.21 and 4.22. Since we already know the harmonic distortion coefficients (γ_1, γ_2 and γ_3) and the noise power ($\int_0^\infty N_\gamma(f)df$) of the ADC1, we can use this information to find the harmonic distortion coefficients and the noise power of the DACN.

4.1.4 Simulation Results

The method described in this section was applied to a 14-bit DAC and a 14-bit Sample and Hold ADC with MATLAB simulation. The ADC and the DAC are modeled as shown in Figure 4.4. The ADC model is divided into two blocks: the first block models the dynamic nonlinearity of the ADC which is represented as $h_{adc}(x)$, and the second block models the quantization process which is represented as $q(x)$. Also, the ADC is assumed to be noisy, and white Gaussian noise, $n_{awgn}(t)$, is added to the output of the ADC model. Thus, the output noise $N_{adc}(t)$ consists of the Gaussian noise, $n_{awgn}(t)$, and

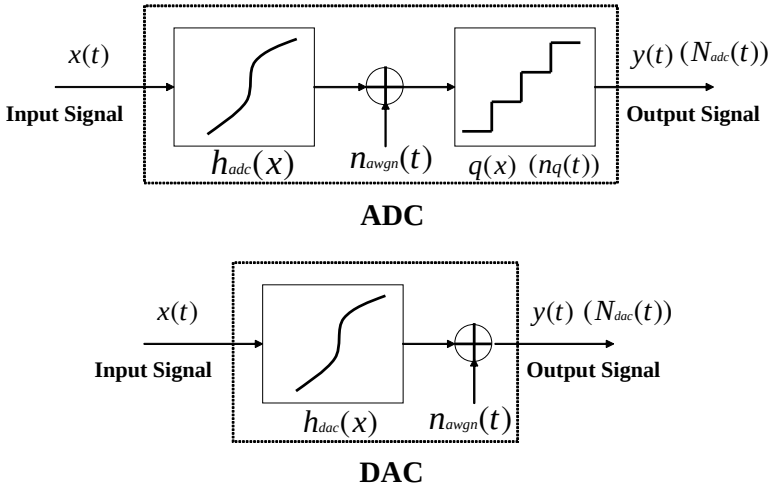


Figure 4.4: Nonlinear ADC and DAC Model

the quantization noise, $n_q(t)$, generated during the quantization process. The DAC is modeled similar to the ADC, except that there is no quantization block in the DAC model.

For simulation, 100 ensembles of the ADC and the DAC models were generated by introducing statistical variations with a Gaussian distribution in the parameters of nonlinear functions, $h_{adc}(x)$ and $h_{dac}(x)$, and power of the additive noise, $n_{awgn}(t)$ described in Figure 4.4. Two sets of ADC and DAC pairs were used to set up the parallel loopback scheme as shown in Figure 4.1 and three performance parameters (SNR, THD and SNDR) were measured using the proposed algorithm. Figure 4.5 to 4.7 shows the plots of the predicted versus the actual values of each DUT performance parameter.

Table 4.1 summarizes the mean and standard deviation of errors between the predicted values and the actual values of performance parameters.

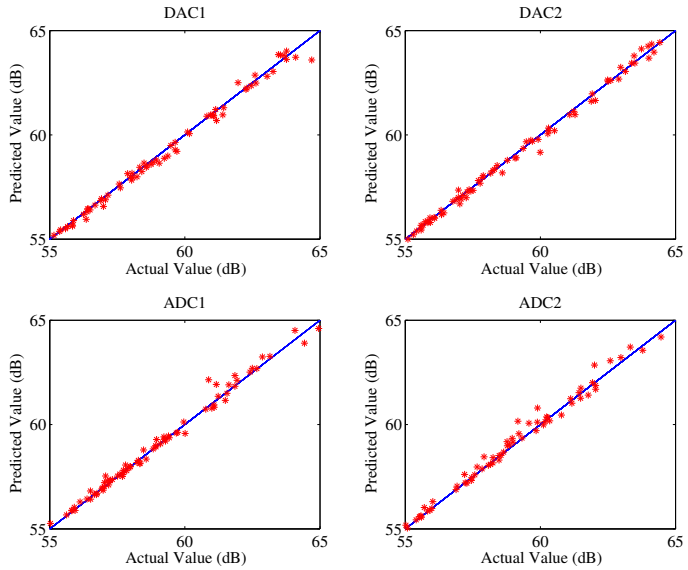


Figure 4.5: Comparison of Actual and Predicted Performance Parameters (SNR)

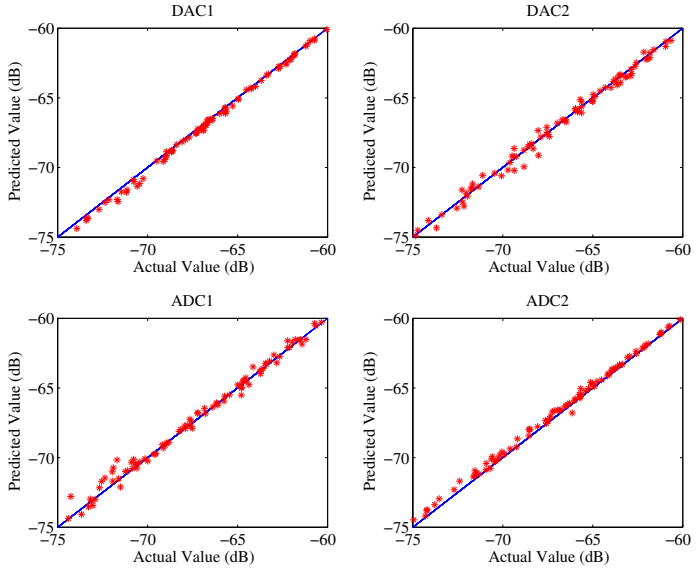


Figure 4.6: Comparison of Actual and Predicted Performance Parameters (THD)

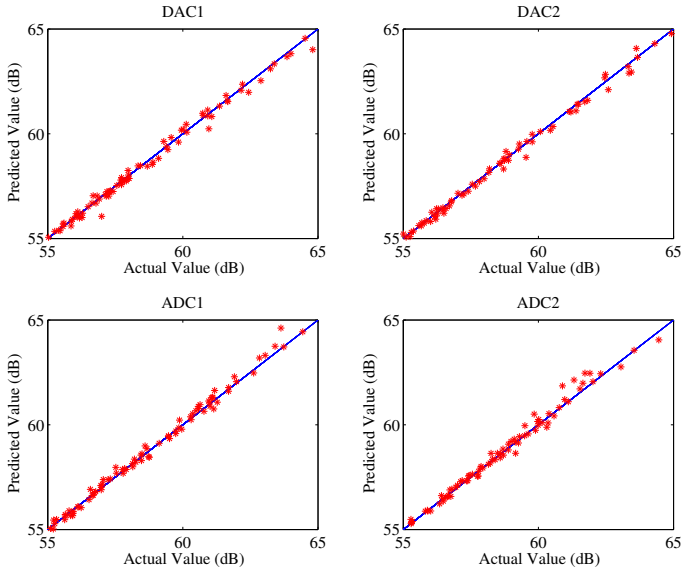


Figure 4.7: Comparison of Actual and Predicted Performance Parameters (SNDR)

It can be seen from the results that prediction errors were less than 2dB in all cases.

Next, the proposed algorithm is applied to the case where there are 10 DUTs (5 ADCs and 5 DACs). Figure 4.8 shows the mean and the standard deviation of prediction errors. We can see that the prediction errors did not increase compared to the case where there were 4 DUTs and this means that the proposed algorithm works well with the increased number of DUTs. It can also be seen that there was no considerable variation in estimation errors among the different DUTs, and this shows that the test order among different DUTs does not affect the test accuracy.

Finally, the resolutions of the DACs and ADCs were varied to see how

Table 4.1: **Mean and Standard Deviation of Performance Parameter Prediction Error**

Parameter		DAC1	DAC2	ADC1	ADC2
SNR	Mean	0.37dB	0.38dB	0.41dB	0.45dB
	STD	0.65dB	1.13dB	1.07dB	0.90dB
THD	Mean	0.31dB	0.32dB	0.40dB	0.34dB
	STD	0.40dB	0.48dB	0.60dB	0.43dB
SNDR	Mean	0.21dB	0.19dB	0.19dB	0.24dB
	STD	0.37dB	0.31dB	0.29dB	0.35dB

these variations affect the test accuracy. This is important since the resolution of the loopback response is limited by the resolution of data converters and this can affect the test accuracy. The resolutions were changed from 10 bits to 14 bits, and the performance parameters were measured in each case. A test setup with two DACs and two ADCs was used for this simulation. Table 4.2 summarizes the mean and standard deviation of prediction errors in various cases. The values in Table 4.2 are averaged values across the four DUTs (two ADCs and two DACs). The results indicate that prediction errors are less than 3dB while maximum error occurred when the DAC had 10 bit resolution. It also indicates that the resolution of the DAC is more critical for the accuracy than the resolution of the ADC.

4.1.5 Summary

In Section 4.1, an efficient parallel test methodology for mixed-signal circuits has been discussed. The algorithm presented in this section can be used to characterize performance parameters of DUTs accurately using digital

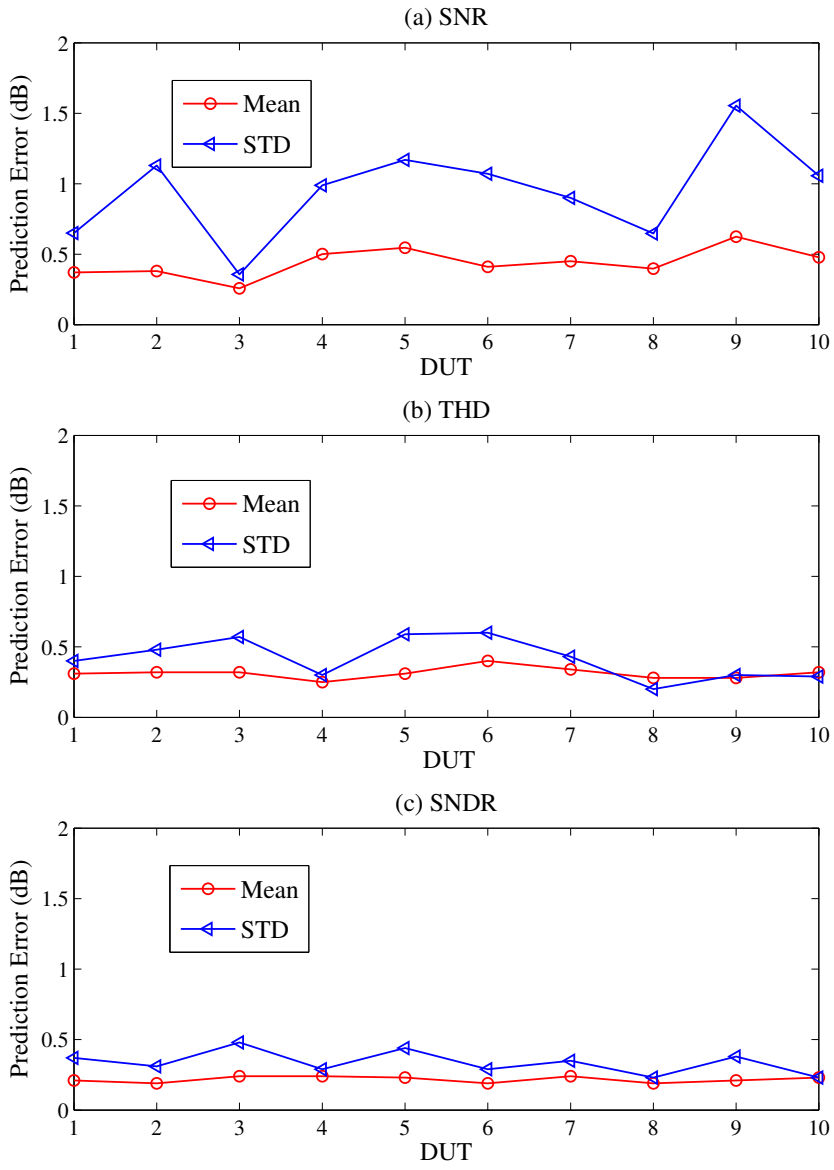


Figure 4.8: Prediction Error of Testing 10 DUTs in Parallel

Table 4.2: **Prediction Error of Parallel Test in Various ADC/DAC Resolution**

Resolution		**SNR**		**THD**		**SNDR**	
ADC	DAC	Mean	STD	Mean	STD	Mean	STD
12bit	14bit	0.35dB	0.73dB	0.34dB	0.50dB	0.16dB	0.20dB
14bit	12bit	0.57dB	0.79dB	0.38dB	0.55dB	0.34dB	0.39dB
12bit	12bit	0.60dB	0.98dB	0.39dB	0.54dB	0.36dB	0.46dB
10bit	14bit	0.30dB	0.55dB	0.45dB	0.70dB	0.19dB	0.28dB
14bit	10bit	0.92dB	1.94dB	0.87dB	1.07dB	0.79dB	1.06dB
10bit	10bit	0.90dB	1.41dB	0.90dB	1.19dB	0.77dB	1.00dB
10bit	12bit	0.51dB	0.61dB	0.35dB	0.38dB	0.29dB	0.27dB
12bit	10bit	1.16dB	1.71dB	0.88dB	0.92dB	0.89dB	1.11dB

test equipments and a DUT board shared among multiple DUTs. A single tone digital sine wave is applied to the DUTs in loopback mode and the resulting digital output response is used to characterize the performance of each DUT separately. The DUT board which contains a simple analog adder and an RMS detector is used to estimate the performance parameters of the multiple DUTs accurately without being affected by the fault masking problem. The presented parallel loopback test algorithm does not depend on the types of DUT being tested, and can thus be applied to general mixed-signal circuits to help reduce the test cost and time. Mathematical derivations and simulation results show the validity of the algorithm.

4.2 Parallel Test of AMS Circuits using Pseudorandom Signals

This presents the parallel test method for analog and mixed-signal circuits using a pseudorandom signal. The contribution of the work presented

in this section is to develop an efficient parallel test method which can characterize the performance parameters of multiple DUTs simultaneously using a common test setups and reduced number of test measurements. This has been achieved by exploiting the properties of the pseudorandom signal which exhibits spread-spectrum characteristics.

Section 4.2 is organized as follows; Section 4.2.1 reviews important properties of the pseudorandom signal which are used to develop the proposed method and Section 4.2.2 presents the parallel pseudorandom test method, detailing its topology and mathematical basis. In Section 4.3.3, the experimental results are presented, and the summary of this research is presented in Section 4.3.4.

4.2.1 Spread-spectrum Properties of Pseudorandom Signal

A pseudorandom signal generated from a LFSR has the property of being *spread-spectrum* which means that a spectrum power of the pseudorandom signal is spread over wide range of frequencies. One important characteristics of the spread-spectrum (or pseudorandom) signal is that it appears to be a random noise in the frequency band it spans. Thus, it can be transmitted with other signals that occupy the same frequency bandwidth with minimum influence on the error rate of those signals. Due to this property, the spread-spectrum signal has been widely used as a channel coding scheme in multiuser communication systems to increase the signal bandwidth per user while not increasing the signal power [41]. Another important characteristic of the

spread-spectrum signal is that it has an i.i.d. (independent and identically-distributed) property which means that the different pseudorandom sequences generated using different LFSR seed values are statistically independent to each other.

The properties described above allow a number of distinct pseudorandom signals which span a common frequency band and a common time slot to be combined together while preserving their individual property. The information embedded in each pseudorandom signal can be recovered from a combined signal using a cross-correlation method. For example, let us assume that N number of pseudorandom signals, which can be generated using the different LFSR seed values, are combined together to generate a composite signal, $x_c(t)$, as follows.

$$x_c(t) = \sum_{k=1}^{N} f_{LFSR}(s_k) = \sum_{k=1}^{N} x_k(t) \tag{4.23}$$

where f_{LFSR} represents the LFSR function that generates the pseudorandom signals and s_k represents the k-th seed value. Thus, the pseudorandom signals ($x_k(t)$) shown in Equation 4.23 are generated by the same LFSR but different seed values.

To extract the information of particular pseudorandom signal ($x_1(t) = f_{LFSR}(s_1)$) from $x_c(t)$, the correlation of $x_c(t)$ and $x_1(t)$ can be used as follows.

$$E[x_c(t)x_1(t-n)] = E[x_1(t)x_1(t-n)] + E[\sum_{k=2}^{N} x_k(t)x_1(t-n)] \tag{4.24}$$

In Equation 4.24, $E[\sum_{k=2}^{N} x_k(t)x_1(t-n)]$ becomes zero since $x_k(t)$ for different k are i.i.d. random processes as explained previously. Thus, we can recover

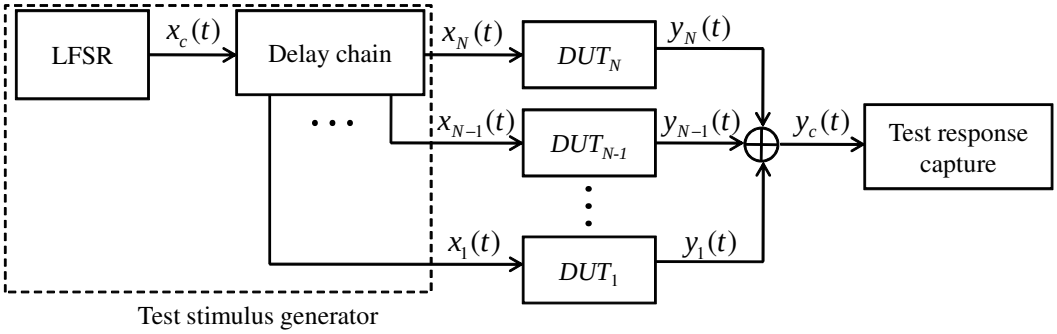


Figure 4.9: Parallel Pseudorandom Test Setup

the information of $x_1(t)$ using the cross-correlation as follows.

$$\tilde{x}_1(n) = \frac{1}{\sigma_x^2} E[x_c(t)x_1(t-n)] \tag{4.25}$$

Section 4.2.2 explains how the spread-spectrum properties of the pseudorandom signal presented so far can be applied to characterize the performance of multiple nonlinear AMS devices in parallel.

4.2.2 Parallel Pseudorandom Test

Figure 4.9 shows the proposed setup for the parallel pseudorandom test in which N DUTs are tested simultaneously. The goal is to extract the information of each DUT's performance separately from the composite output, $y_c(t)$ shown in Equation 4.9. To achieve this goal, the spread-spectrum properties of the pseudorandom signal explained Section 4.2.1 and the pseudorandom test method described in Section 3.4 are used to develop the efficient performance characterization algorithm. Additionally, the Volterra series model described

in Section 3.2 is used to represent the nonlinear behaviors of the DUTs accurately.

As shown in Figure 4.9, a test stimulus generator and test response capturing equipment are shared among multiple DUTs. We can see that the test pattern generator consists of a LFSR and a delay chain. In Section 4.2.1, it is explained that the pseudorandom sequences generated using different LFSR seed values are uncorrelated to each other. In the proposed parallel pseudorandom test method, instead of using different LFSR seed values to generate multiple pseudorandom sequences, the delay chain is used to generate the pseudorandom signals that are uncorrelated to each other. This reduces the hardware complexity of the test stimulus generator, and makes it possible to feed the test stimuli to multiple DUTs at the same time using the single test stimulus generator. Each pseudorandom signal generated from the test stimulus generator ($x_1(t)$ through $x_N(t)$ shown in Figure 4.9) is fed to a different DUT, and the resulting outputs from each DUT are added together to generate the composite output signal $y_c(t)$. Using the Volterra series model, $y_c(t)$ can be expressed as follows.

$$\begin{aligned} y_c(t) = \sum_{k=1}^{N} \{ & \sum_{\tau} h_{k1}(\tau) x(t-\tau) + \sum_{\tau_1} \sum_{\tau_2} h_{k2}(\tau_1, \tau_2) x(t-\tau_1) x(t-\tau_2) \\ & + \sum_{\tau_1} \sum_{\tau_2} \sum_{\tau_3} h_{k3}(\tau_1, \tau_2, \tau_3) x(t-\tau_1) x(t-\tau_2) x(t-\tau_3) \} \end{aligned} \tag{4.26}$$

where h_{k1}, h_{k2} and h_{k3} represent the first, second and third Volterra kernels of the kth DUT (DUT_k) respectively. In the rest of this section, algorithms to identify the Volterra kernels of DUT_1 from Equation 4.26 will be discussed.

Please note that the same method can be applied to other DUTs without modification.

In order to extract the information of odd-order Volterra kernels (i.e., the first-order and the third-order kernels) and even-order Volterra kernel (i.e., the second-order kernel) from $y_c(t)$, two types of the *pattern combination* can be used as follows.

$$\begin{aligned} x^o_{pc1}(t) &= x_1(t)x_1(t+n)x_1(t+p) \\ x^e_{pc1}(t) &= x_1(t)x_1(t+n) \end{aligned} \tag{4.27}$$

where $x_1(t)$ is the pseudorandom signal applied to the DUT_1 as shown in Figure 4.9. First, to identify the odd-order Volterra kernels, $x^o_{pc1}(t)$ can be used to calculate the cross-correlation as follows.

$$\begin{aligned} R^o_{yx1}(m) =& E[y_c(t)x^o_{pc1}(t-m)] \\ =& \sum_{\tau} h_{11}(\tau)\mu^o_4 + \sum_{\tau_1}\sum_{\tau_2}\sum_{\tau_3} h_{13}(\tau_1,\tau_2,\tau_3)\mu^o_6 \\ &+ \sum_{k=2}^{N}\{\sum_{\tau} h_{k1}(\tau)\mu^o_{k4} + \sum_{\tau_1}\sum_{\tau_2}\sum_{\tau_3} h_{k3}(\tau_1,\tau_2,\tau_3)\mu^o_{k6}\} \end{aligned} \tag{4.28}$$

where μ^o_4 and μ^o_6 are the fourth-order and the sixth-order moments of $x_1(t)$ which can be expressed as Equation 3.33. Moreover, μ^o_{k4} and μ^o_{k6} are the fourth-order and the sixth-order moments which composed of $x_1(t)$ and $x_k(t)$ as follows.

$$\begin{aligned} \mu^o_{k4} &= E[x_k(t-\tau)x_1(t-m)x_1(t+n-m)x_1(t+p-m)] \\ \mu^o_{k6} &= E[x_k(t-\tau_1)x_k(t-\tau_2)x_k(t-\tau_3)x_1(t-m)x_1(t+n-m)x_1(t+p-m)] \end{aligned} \tag{4.29}$$

Since the pseudorandom signals are i.i.d. random processes as explained in the previous section, μ^o_{k4} and μ^o_{k6} become zero for all k other than $k = 1$. Then,

we can simplify Equation 4.28 as follows.

$$\begin{aligned} R^{o}_{yx1}(m) &= E[y_c(t)x^{o}_{pc1}(t-m)] \\ &= \sum_{\tau} h_{11}(\tau)\mu^{o}_{4} + \sum_{\tau_1}\sum_{\tau_2}\sum_{\tau_3} h_{13}(\tau_1,\tau_2,\tau_3)\mu^{o}_{6} \end{aligned} \tag{4.30}$$

We can see that Equation 4.30 is in the same form as Equation 3.32. Hence, the Volterra kernel identification method described in Section 3.4.2.2 can be used to find the odd-order Volterra kernels of the DUT_1 from Equation 4.30.

Next, we can use following cross-correlation equation to find the even-order Volterra kernel of the DUT_1.

$$\begin{aligned} R^{e}_{yx1}(m) &= E[y_c(t)x^{e}_{pc1}(t-m)] \\ &= \sum_{\tau_1}\sum_{\tau_2} h_{12}(\tau_1,\tau_2)\mu^{e}_{4} + \sum_{k=2}^{N}\{\sum_{\tau_1}\sum_{\tau_2} h_{k2}(\tau_1,\tau_2)\mu^{e}_{k4}\} \end{aligned} \tag{4.31}$$

where μ^{e}_{4} is the fourth-order moment of $x_1(t)$ which can be expressed as Equation 3.27 and μ^{e}_{k4} is the fourth-order moment which composed of $x_1(t)$ and $x_k(t)$ as follows.

$$\mu^{e}_{k4} = E[x_k(t-\tau_1)x_k(t-\tau_2)x_1(t-m)x_1(t+n-m)] \tag{4.32}$$

To identify the values of the second-order Volterra kernel, h_{12}, two different cross-correlation calculations are required listed as below.

$Case1$: If $n \neq 0$ in Equation 4.32, μ^{e}_{k4} becomes zero for $k \geq 2$, because in this case four pseudorandom sequences shown in Equation 4.32 are i.i.d. random processes. Then, following the approach shown in Section 3.4.2.1, Equation 4.31 can be expressed as follows.

$$R^{e(i)}_{yx1}(m;n) = 2\sigma^{4}_{x}h_{12}(m,m-n) \tag{4.33}$$

$Case2$: If $n = 0$, the value of μ^e_{k4} can be expressed as follows.

$$\mu^e_{k4} = \begin{cases} 0 & (\text{if } \tau_1 \neq \tau_2) \\ \sigma_x^4 & (\text{if } \tau_1 = \tau_2) \end{cases} \tag{4.34}$$

Then, Equation 4.31 can be expressed as follows.

$$R^{e(ii)}_{yx1}(m) = \sigma_x^4 h_{12}(m, m) + N\sigma_x^2 E[y_c(t)] \tag{4.35}$$

Using Equations 4.33 and 4.35, we can identify the second-order Volterra kernel of the DUT_1 as follows.

$$\tilde{h}_{12}(m, m-n) = \begin{cases} \dfrac{R^{e(i)}_{yx}(m;n)}{2\sigma_x^4} & (\text{if } n \neq 0) \\ \dfrac{R^{e(ii)}_{yx}(m) - N\sigma_x^2 E[y_c(t)]}{2\sigma_x^4} & (\text{if } n = 0) \end{cases} \tag{4.36}$$

Now, the performance of DUT_1 can be separately characterized from the composite response $y_c(t)$ using Equations 4.30 and 4.36.

4.2.3 Experimental Results

In the following section, simulation results and hardware measurement results are presented to validate the presented parallel pseudorandom test method.

4.2.4 Simulation Results

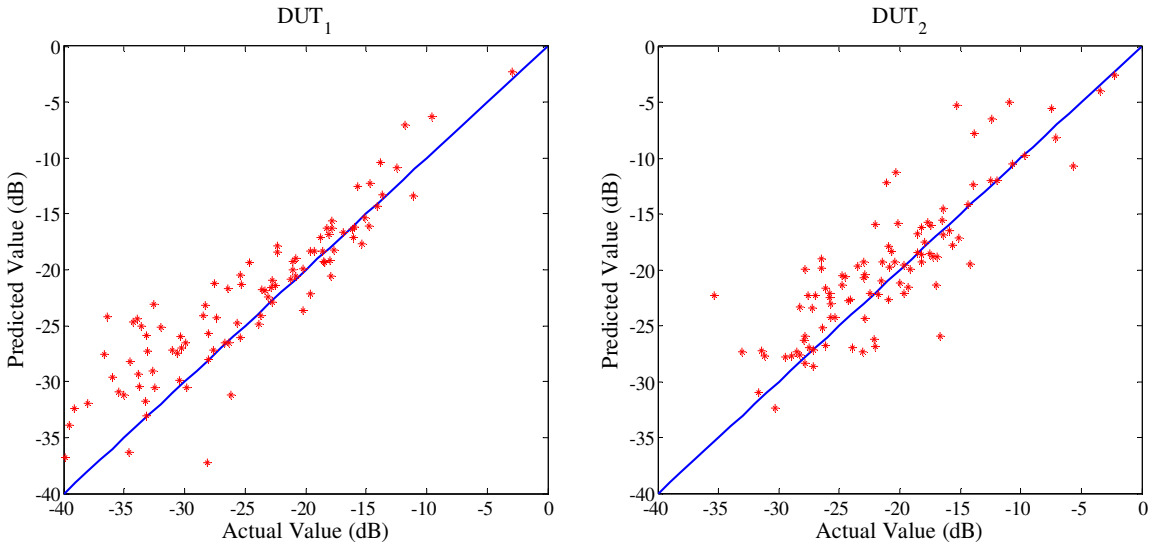
The method described above was applied to characterize the performance of 12-bit DACs with MATLAB simulation. The DAC model used in this simulation was same as the DAC model shown in Figure 4.4 in which

Table 4.3: **Mean and Standard Deviation of Performance Parameter (THD) Prediction Error**

Device Under Test	Mean	Standard Deviation
DUT_1	3.12dB	2.90dB
DUT_2	3.06dB	3.27dB
DUT_3	3.04dB	2.42dB
DUT_4	3.74dB	2.76dB

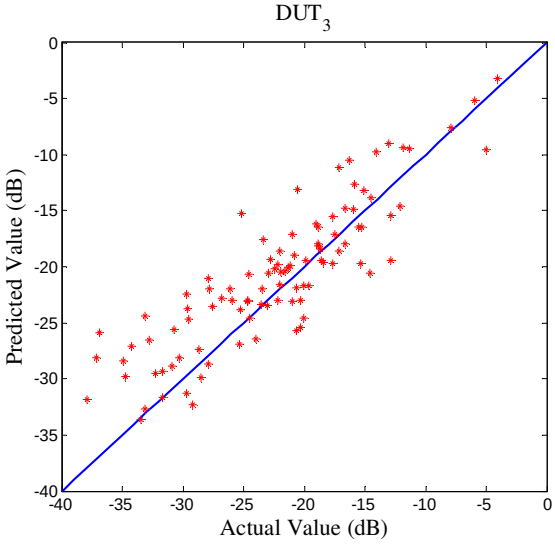
the nonlinearity of the DAC model is represented as $h_{dac}(x)$, and the white Gaussian noise, $n_{awgn}(t)$, is added to the output response of the DAC. Four instantiations of the DAC model were used to set up the parallel pseudorandom test environment (i.e., $N = 4$ in Figure 4.9), and the THD values of each DAC were measured using the proposed method. Moreover, the amounts of delays between the pseudorandom sequences generated from the test stimulus generator were same as ten clock cycles in this simulation. (i.e., $x_k(n) = x_{k-1}(n-10)$ for $k = 2, 3, 4$)

For simulation, 100 ensembles of the DAC model were generated for each DUT (DUT_1 through DUT_4) by introducing statistical variations with a Gaussian distribution in parameters of nonlinear functions, $h_{dac}(x)$, and power of the additive noise, $n_{awgn}(t)$ described in Figure 4.4. Figure 4.10 shows the plots of the predicted versus the actual values of the THD for each DUT. Also, Table 4.3 summarizes the mean and standard deviation of error in predicting the THD values.

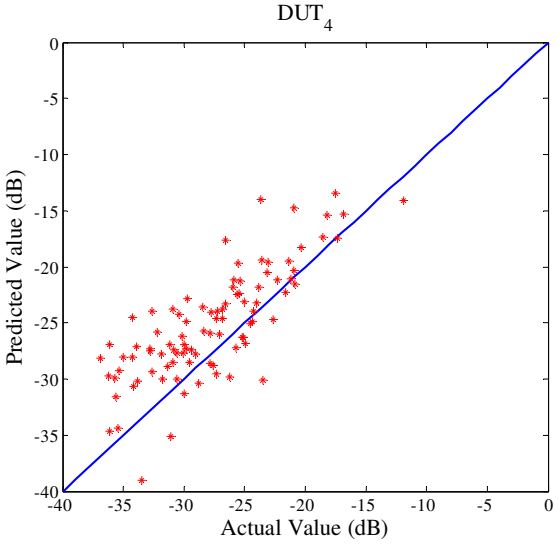


(a) THD of DUT_1

(b) THD of DUT_2

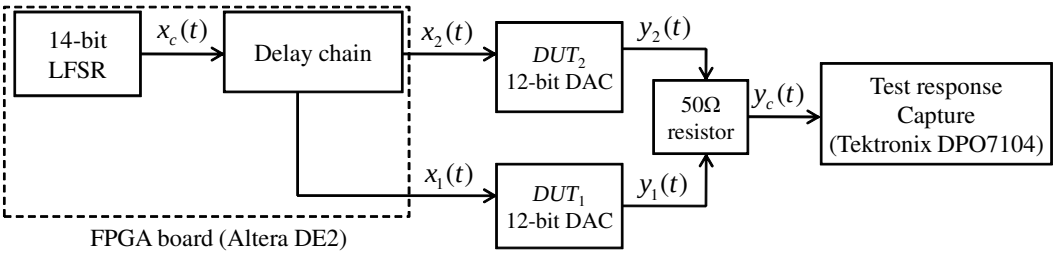


(c) THD of DUT_3

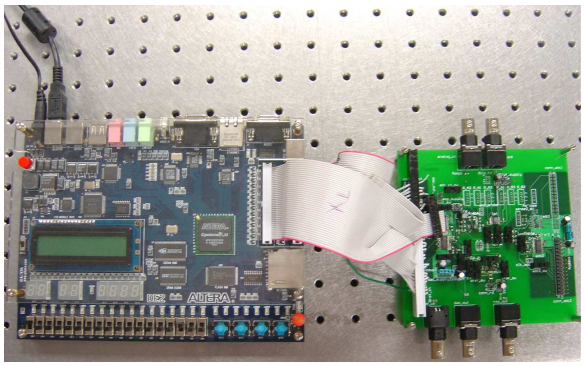


(d) THD of DUT_4

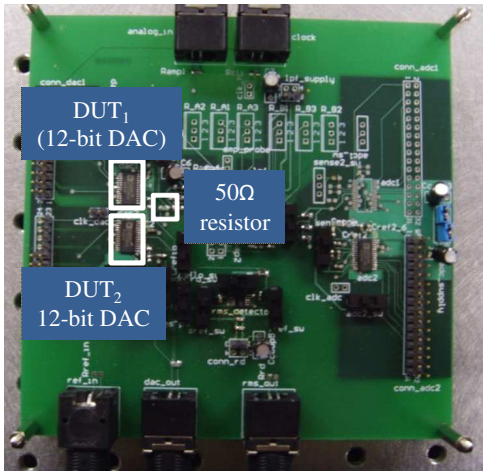
Figure 4.10: Comparison of Actual and Predicted Values of THD



(a) Block Diagram of Measurement Setup



(b) FPGA board and DUT board



(c) Configuration of the DUT board

Figure 4.11: Hardware Measurement Setup

4.2.5 Hardware Measurement Results

To verify the presented parallel pseudorandom test method in practical test environment, the method was applied to characterize the performance of 12-bit current-steering DACs from Analog Devices™ [1]. Figure 4.11 shows the measurement setup. The LFSR was built using a 14-stage maximum-length sequence generator and phase shifters as shown in Figure 3.10. In Figure 4.11, the delay between two pseudorandom sequences ($x_1(n)$ and $x_2(n)$) was equal to ten clock cycles. (i.e., $x_2(n) = x_1(n-10)$) In this measurement, two DACs were tested simultaneously. The outputs from these DACs were combined using a 50Ω resistor, and the combined output was captured by a digital oscilloscope as shown in Figure 4.11. The test and validation flow for this measurement was similar to the flow shown in Figure 3.11, while in this measurement, the THD values of two DACs were predicted using the parallel pseudorandom test method, and compared to the actual values measured using the conventional test method.

Table 4.4 summarizes the predicted and the actual values of the THD for various cases with different input frequencies. In Table 4.4, THD_1 and THD_2 represent the THD values of the DUT_1 and the DUT_2 respectively. As we can see from the table, the predicted and the actual values are well correlated for most cases, and the prediction errors are comparable to the single DUT test case shown in Section 3.4.3. All the prediction errors are held within 2.8dB of the actual values for the DUTs. To calculate the specification parameters shown in Table 4.4, the parallel pseudorandom test method required a

Table 4.4: Hardware Measurement Data - the DUT operating at 25MHz

Input Frequency	Input Amplitude	Specification	Actual value (dB)	Predicted value (dB)	Error (dB)
50kHz	0dBFS	THD_1	-30.82	-28.10	2.72
		THD_2	-29.20	-29.00	0.20
100kHz	0dBFS	THD_1	-28.69	-29.40	0.71
		THD_2	-29.13	-28.29	0.84
250kHz	0dBFS	THD_1	-29.73	-29.06	0.67
		THD_2	-29.38	-28.12	1.26
500kHz	0dBFS	THD_1	-30.89	-29.23	1.66
		THD_2	-29.54	-28.19	1.35

single measurement while the conventional method required separate measurements for each case which results in eight measurements each with different sinusoidal inputs.

4.2.6 Summary

Section 4.2 presents an efficient pseudorandom test method which is aimed at characterizing the performance of multiple DUTs simultaneously. The method presented in this section uses i.i.d. sequences of pseudorandom signals, which can be generated using simple LFSR and a delay chain, to excite multiple DUTs at the same time. The resulting test responses from each DUT are added together to generate the composite test response which can be captured using a single test equipment. The specification parameters of each DUT are calculated separately using the spectral characteristics of the pseudorandom signal. The parallel pseudorandom test method makes it possible to test multiple DUTs using a single test setup with minimum

overhead of extra test circuitry, and thus reduces the test time as well as the test cost considerably. Experimental results exhibit good correlations with the theory, and thus validate the efficiency of the proposed method.

Chapter 5

At-speed Test of High-speed DUT using Reconfigurable Built-off Test Interface

This chapter presents a flexible test structure which uses low-speed ATE in testing high-speed DUT. The goal of this work is to enable the low-cost ATE to actively control the at-speed test of high-speed DUTs, and monitor the test procedure closely. The built-off test interface (BOTI) circuit has been developed to achieve this goal.

As described in the Chapter 2, one of the main reasons for increasing the ATE cost is the difficulty in sending high-speed signals along the long data channels that lie between the ATE and the DUT. Figure 5.1 shows the typical setup to test semiconductor devices in production using the ATE. The signals communicated between the *tester* and the DUT should go through *channel A* and *channel B*, and in most cases, the length of the *channel A* is much longer than that of the *channel B*. The idea of using the built-off test interface is to implement the test interface on the DUT board (such as WMB or PCB) and use this interface to deal with the high-speed signals on the short *channel B* while the tester handles only the low-speed signal to communicate with the test interface on the long *channel B*. Under the test framework developed here,

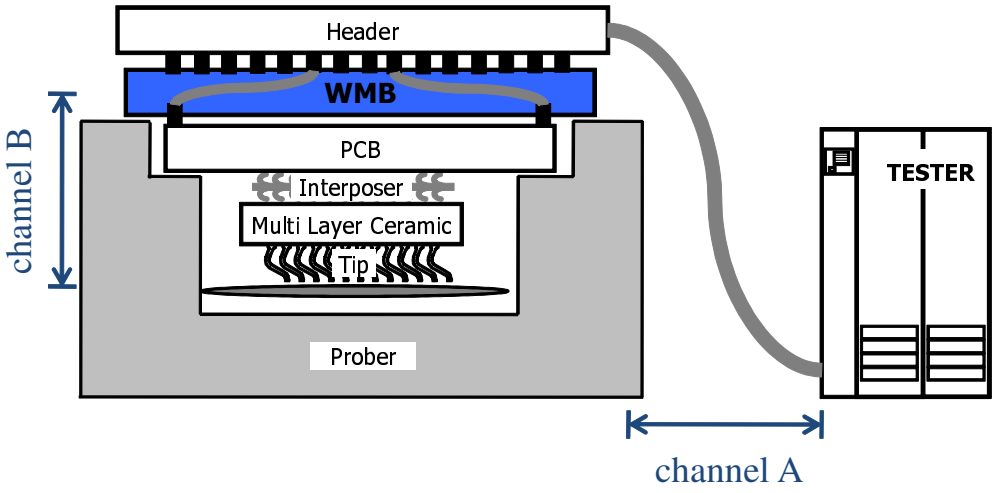


Figure 5.1: Practical Test Setup using the ATE

the ATE and the DUT are communicating through the BOTI module which consists of reconfigurable Finite-State Machine (FSM) and Phase-Locked Loop (PLL) to support seamless signal communication between two sides operating at different frequencies. The low-cost ATE maintains the control of the high-speed test procedures with the help of the BOTI module, and this makes the proposed test method flexible and scalable to various applications. Also, in order to maintain reliable off-chip signal communication, the off-chip channel skew is measured and compensated on the BOTI module using an on-chip Time-Domain Reflectometer (TDR) measurement circuit [18].

5.1 Architecture and Functionality of BOTI

As described in Figure 5.2, the BOTI module consists of three sub-modules which are BOTI Controller, Clock Generator Module and I/O Skew

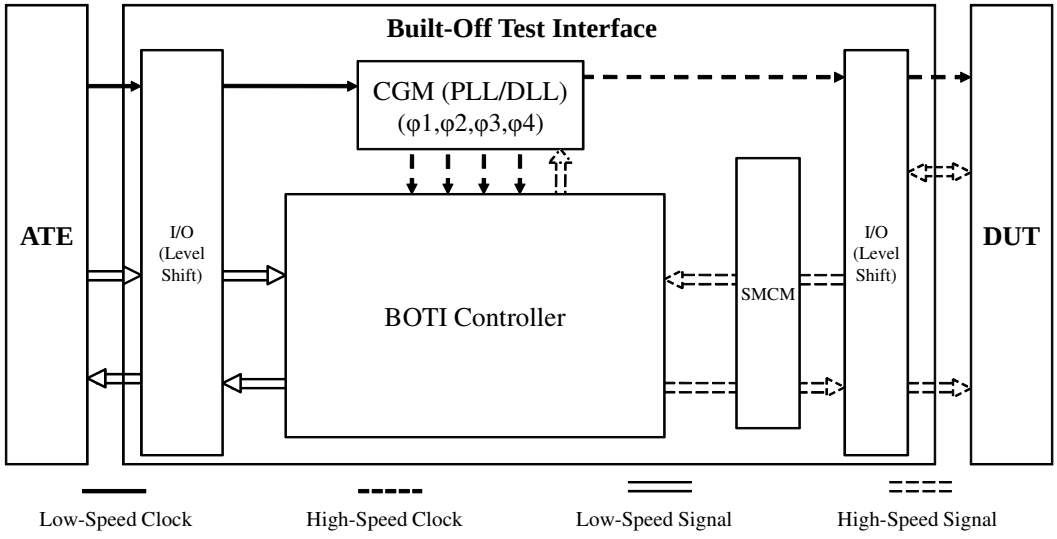


Figure 5.2: Block Diagram of BOTI Module

Measurement & Compensation Module. The functionalities of each module are explained in the following.

5.1.1 Clock Generator Module

As will be described later in this chapter, the BOTI requires high-speed clock signals with several different phases in order to maintain accurate test pattern generation and data read operations. The main goal of the *Clock Generator Module* (CGM) is to generate these clock signals accurately using control signals coming from the ATE. To achieve this goal, the CGM consists of a Phase-Locked Loop (PLL) and a Delay-Locked Loop (DLL) as shown in Figure 5.3. The PLL synthesizes the high-speed clock signal using the low-speed clock signal coming from the ATE as the reference, and feeds the synthesized

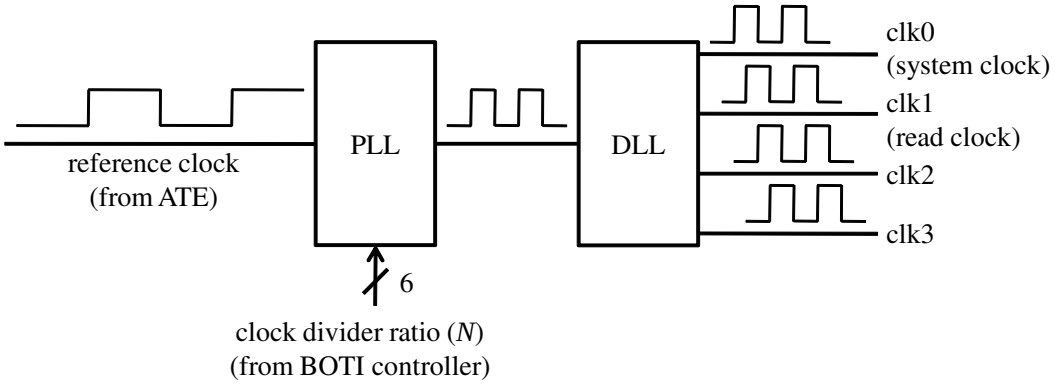


Figure 5.3: Block Diagram of CGM

clock to the DLL. The frequency of synthesized clock can be modified on the fly by controlling the clock divider, and a 6-bit register inside the CGM is used to control the clock multiplication ratio, N, as shown in Figure 5.3. The DLL can generate M number of output clock signals whose clock frequencies are same as the clock input to the DLL, but has different phases. Thus, the phase difference between each clock is $2\pi/M$. The BOTI method explained in this dissertation uses four clock signals with phase difference of 90 degree.

5.1.2 Skew Measurement & Compensation Module

In the proposed test framework, the BOTI circuit is implemented off-chip, and thus test patterns have to go through the off-chip environment which will increase the signal skew among different I/O channels. To compensate this skew, in conventional test setup, the ATE measures the TDR for each channel and adjusts timings of launching signals based on the results of the

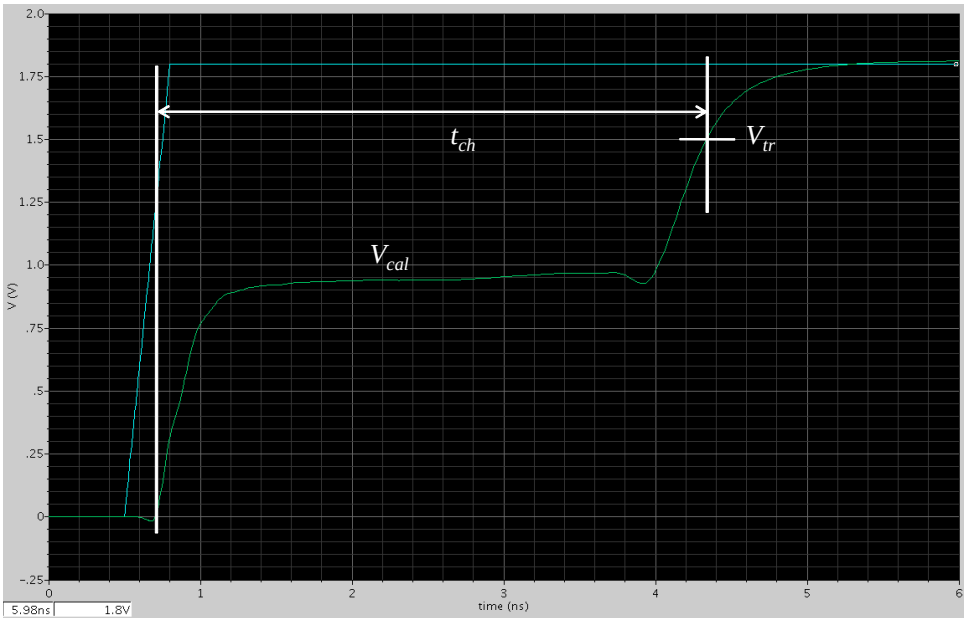


Figure 5.4: Skew Measurement Signal

TDR measurements. Since the BOTI module is placed in between the ATE and the DUT, the BOTI module should measure the off-chip signal skew by itself, and this task is done by the *Skew Measurement & Compensation Module* (SMCM) [18].

The SMCM is implemented on every I/O pad which communicates with the DUT. At first, the SMCM launches a *calibration signal* to selected I/O channels, and this signal is reflected at the other end of each I/O channel due to a termination load. Figure 5.4 shows an example of the calibration signal. At first when the calibration signal is launched, the signal level rises to V_{cal} and stays flat until the reflected signal arrives. Once the reflected signal arrives, the signal level starts to rise to I/O supply voltage level. Then, the

SMCM measures the t_{ch} (the time required to reach the trigger voltage level (V_{tr}) after launching the calibration signal) using a Vernier delay line, and the measured value is saved in registers in each SMCM which called *skew registers* in this dissertation.

When test patterns are launched to the DUT in normal operation mode, they go through programmable delay line in each compensation module whose delay is controlled by the value stored in the skew registers. Same adjustment happens when the test response signals launched by the DUT are received at the BOTI module. In a pre-silicon design stage, the maximum resolution of the skew measurement and the compensation of the SMCM was set to 50ps. Due to the inevitable process variation of the CMOS devices used in the SMCM, the actual resolution can be worse than 50ps, and the detailed measurement results are presented in the section 5.3.

5.1.3 BOTI Controller

The *BOTI Controller* (BOTC) has two objectives: the first is to decode the test instructions coming from the ATE (*ATE instructions*) and generate internal instructions that will be used inside the BOTI (*BOTI instructions*), while the second is to control the operation of the BOTI based on the BOTI instructions.

The main issue in decoding the ATE instructions comes from the fact that, since the ATE is operated N times slower than the BOTI (and the DUT), the ATE can only send the instructions to BOTI every N cycles of the BOTI

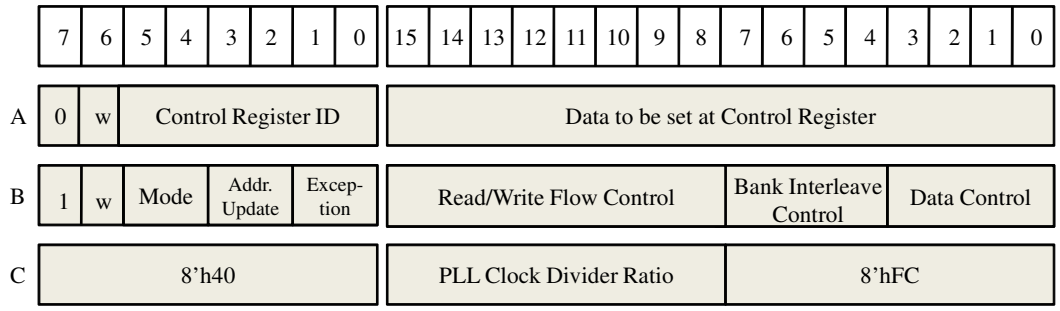


Figure 5.5: Instruction Set for Memory Test

system clock, where N is the multiplication ratio explained in the section 5.1.1. Unlike the self-test method, in which the self-test logic need not consider the difference in operating speed between the ATE and the DUT since it just runs the pre-determined test program from start to end, in the BOTI method, the ATE has the main control over the test program, and thus the operation of the BOTC should be controlled by the ATE instructions through the entire test program. This means that i) the ATE instruction set should contain enough information to run an at-speed test program which is running faster than the update rate of the ATE instructions, and ii) the BOTC should process the low-speed ATE instructions and produce the high-speed BOTI instructions in accordance with the high-speed operation of the BOTI. The first issue is discussed in this section, while the second issue is discussed in the section 5.2.

5.1.3.1 Test Pattern Generation

In many applications that requires at-speed test such as dynamic test of AMS circuit or memory test, the test pattern has high regularity, which means

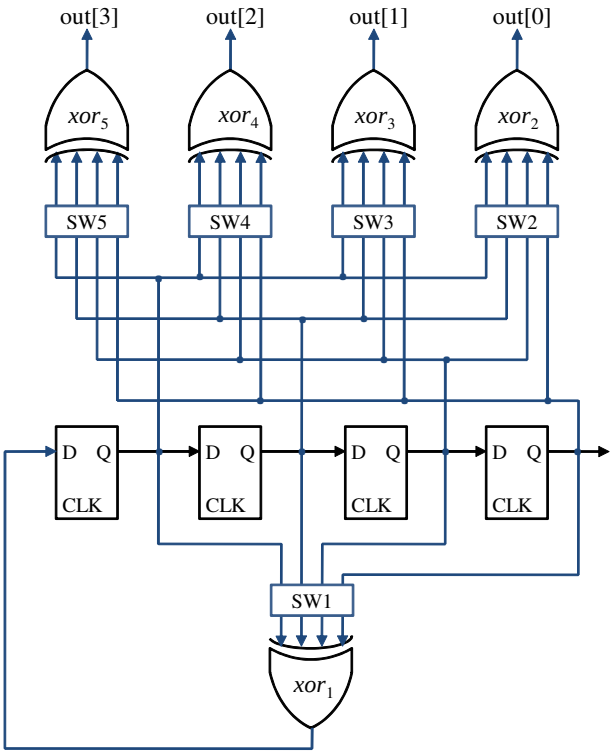
that similar sets of test patterns are repeated for entire test program [58]. In the BOTI method, this regularity is used to compress the information required to run high-speed test program and to implement the compressed information with the low-speed ATE instruction set.

Inside the BOTC, a reconfigurable FSM is implemented to generate the test patterns whose flow is configured by the ATE instructions. Due to the regularity of the test patterns, the configuration of the FSM does not have to be updated frequently, and thus, the ATE instructions can be used to update the FSM flow regularly at a lower rate than the FSM operating speed. The set of signals to be generated in each state of the FSM operation and the intervals between each state are determined by the type of DUT to be tested and the type of test programs to be executed. These FSM operations can be controlled by various registers which called *control registers* in this dissertation. The values of control registers can be set by the ATE instruction before the test sequence begins, and also while the test sequence is running.

The examples of generating a pseudorandom test pattern and a memory March test pattern are presented to illustrate the operations of the FSM and the BOTC. At first, Figure 5.5 shows several examples of ATE instructions and how these instructions can be decoded in the BOTC module to generate memory March test pattern. In this example, the ATE instruction set consists of two words, one with 8-bit and the other with 16-bit. In the March test, whole test sequences can be divided into one of following modes: *read*, *write*, *read/write* and *write/read* mode. Typically, one mode of operation needs to be

run consecutively for multiple cycles while the data to be written to DUT (or expected from DUT) and the address are changing from cycle to cycle. In each mode, the FSM goes through several states of signal generations, and in each state, set of control signals (such as *RAS, CAS, WE and etc.*), data signals and address signals are generated in pre-determined order which is determined by the values stored at the control registers. In this example, the ATE can set the values of the control registers using the instruction set A shown in Figure 5.5, in which the 8-bit word contains the ID number of each control register, while the 16-bit word contains the value to be stored at the control register identified by the 8-bit word. Also, the flow of the data and the address, which require cycle-to-cycle update, can be controlled by the instruction set B.

Next, to generate the pseudorandom patterns using the BOTI module, a maximum-length sequence generator and a phase shifter can be implemented using reconfigurable FSM. Let us explain the operation of the BOTC using an example of 4-bit pseudorandom pattern generator as shown in Figure 5.6(a). In Figure 5.6(a), one of the XOR gate, xor_1 , is used to formulate the specific polynomial equation for the maximum-length sequence generator and the other XOR gates, xor_2 through xor_5, are used as phase shifters. Configurations of these phase shifters and the maximum-length sequence generators can be set using various switches, SW_1 through SW_5, shown in Figure 5.6(a). The configuration information of these switches and the seed value of the LFSR are sent to the BOTI module using the instruction set A shown in Figure 5.6(b) before test begins, or using the instruction set B while the test program is



(a) 4-bit Pseudorandom Pattern Generator

	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A	0	w	Control Register ID						Data to be set at Control Register															
B	1	w	Control Register ID						Data to be set at Control Register															
C	8'h40								PLL Clock Divider Ratio								8'hFC							

(b) Instruction Set for Pseudorandom Pattern Generation

Figure 5.6: Example of Generating Pseudorandom Pattern using BOTI

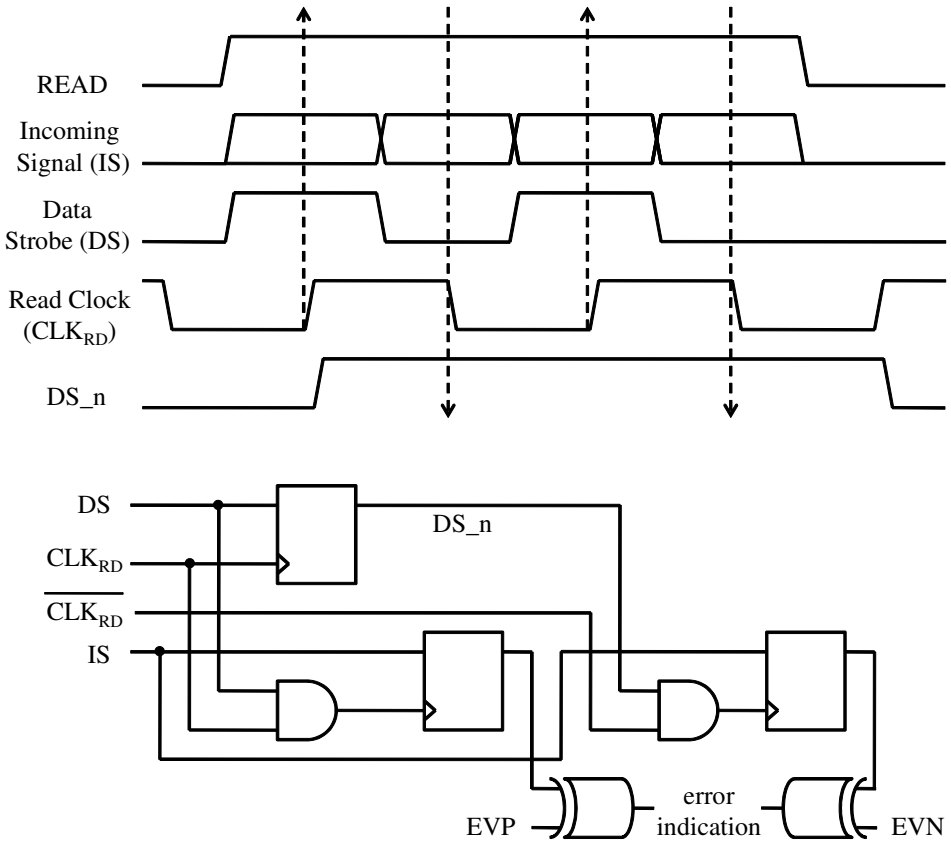


Figure 5.7: Data Read Scheme

currently running. Detailed explanation of instructions decoding scheme is presented in section 5.2.

5.1.3.2 Data Read Scheme

When reading the test responses generated by the DUT, it is important for the BOTI to latch the incoming data at the right timing in spite of off-chip channel skews and hazardous glitches. The test responses, which called *incom-*

ing signals in this dissertation, launched from the DUT to the BOTI suffer from same off-chip environment as the test patterns, which called *outgoing signals* in this dissertation, launched from the BOTI to the DUT. While the channel skew introduced on the outgoing signals can be compensated for by controlling the launching time on the BOTI module, the skew introduced on the incoming signals cannot be completely compensated for since the BOTI does not have the information of the incoming signals' launching time. To resolve this issue and make the read operation reliable, the BOTC uses a delayed clock signal (*read clock*) dedicated to the read operation and the *data strobe* signal. The data strobe signal is commonly used in high-speed circuits to synchronize the signal communication between the sender and the receiver by using a set of pulse signals synchronized with the data signals. The read clock is generated by the CGM and lags the system clock by a quarter cycle as shown in Figure 5.3.

When the incoming signals are expected, the BOTC first asserts the *READ* flag which informs the BOTI to be ready for latching the incoming signals and comparing them against the expected values. Figure 5.7 depicts the signals used in the data read operation. When the *READ* flag is high and the data strobe signal goes high, the BOTI latches the incoming signal on the rising edge of the *read clock*. Some high-speed devices such as Double-Data-Rate DRAM send out the data at both rising and falling edge of the system clock, which means that the BOTI needs to latch the incoming signals at both edges. In order to latch the incoming data properly at the negative edge

of clock, the BOTC produces the enable signal, *DS_n*, using the data strobe signal as shown in Figure 5.7, and use it to control the data capture. This scheme makes sure that the BOTI captures incoming data at the right timing as long as relative skew between the incoming signal and the data strobe signal is less than one-fourth of the DUT clock period. The SMCM can be used to reduce the relative skew within this range. The incoming signal latched during the read operation is compared against the expected value shown as EVP and EVN in Figure 5.7. Then, the BOTC produces an error indication signal if the value of the incoming signal is different from the expected value, and sends the error indication signal to the ATE. The data strobe signal may not be available in some cases. In these cases, the BOTC can still latch the incoming signal safely using the *READ* signal in place of the data strobe signal.

5.1.3.3 Clock Frequency Manipulation

As explained before, the operations of the CGM and the SMCM are controlled by the BOTC, and the one important operation associated with the CGM is to change the clock frequency flexibly. In many at-speed test applications, it is required to manipulate the test clock frequency to examine the DUT performances in several different corners. The BOTC can manipulate the clock frequency by controlling the clock multiplication ratio (N) of the CGM as shown in Figure 5.3, and this ratio can be modified on the fly while the test program is running. When changing the clock frequency while the test program is running, the BOTI clock signal can be temporarily in a metastable

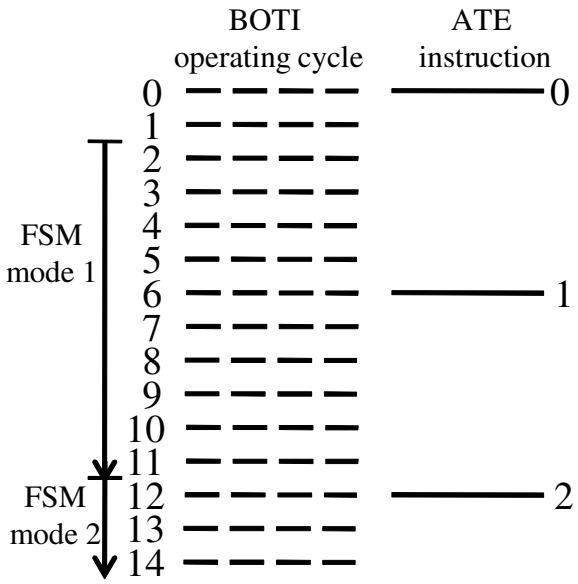


Figure 5.8: High-speed BOTI operation and Low-speed ATE instruction

state which may lead to a loss of information stored in the internal registers of the BOTC. To prevent this from happening, the information stored in the internal registers are copied to respected *shadow registers* which are clocked by the slow ATE clock. Once the BOTI system clock is completely locked to a new frequency, the information saved in the *shadow registers* are copied back to the original registers which are clocked by the BOTI system clock.

5.2 Handshake Procedure

This section presents a handshake procedure between the ATE and the BOTI which is required to ensure the proper communication between two sides, which are operating at different frequencies, without interrupting the

at-speed test procedure.

At first, the example shown in Figure 5.8 explains the issue arising from the fact that the ATE is operating slower than the BOTI. In this example, the BOTI is running six times faster than the ATE. As explained in the previous section, the FSM is operated based on the values stored at the control registers, and these register values are set by the ATE instructions before the test program begins or while the test program is running. As we can see from the example, one mode of FSM operation is running for nine cycles (cycle 2-11) of the fast clock, and then changes to different mode at the *cycle 12*. Since the ATE can send the instruction to the BOTI every six cycles of the fast clock (assuming the ATE can send the instructions only at the positive edge of the slow clock), the ATE should send the instruction to change the FSM mode using the *ATE instruction 1*. However, upon receiving the *ATE instruction 1*, the BOTI should not decode the incoming instruction and should not update the control register values immediately, because this will lead to change of the FSM operation mode earlier than it should be. Thus, proper handshake procedure is required to decode the *ATE instruction 1* and update the control registers at the *cycle 11* of the BOTI operation, so that the FSM operation can be changed to different mode properly at the *cycle 12*.

Figure 5.10 shows timing diagram to illustrate the handshake procedure implemented in the BOTC to solve the issues mentioned above. Also, in order to maintain proper handshake procedure, the BOTC requires both fast clock and slow clock, and these two clock signals should be synchronized with each

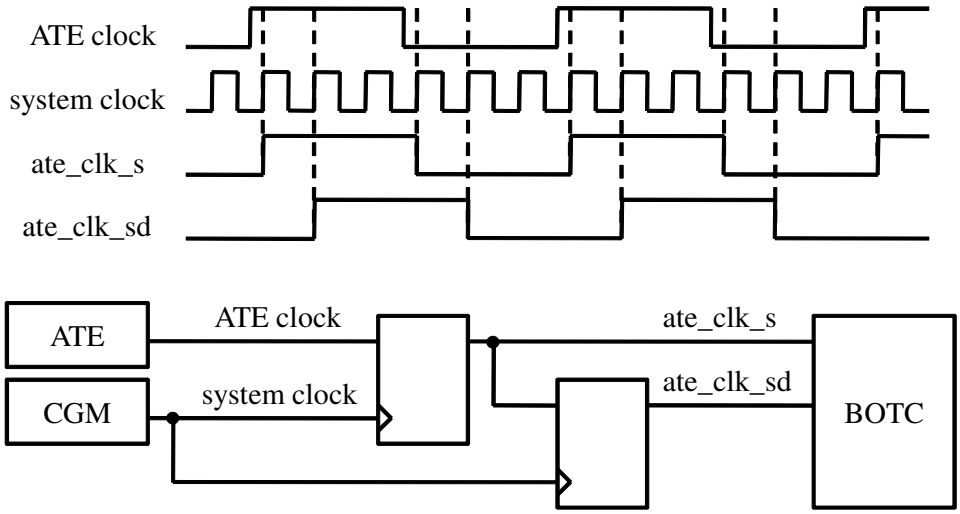


Figure 5.9: Clock Synchronization Scheme

other. To maintain the synchronization between the two clock signals, the BOTC generates synchronous low-speed clock signals (*ate_clk_s* and *ate_clk_sd*) internally which are aligned to the BOTI system clock as shown in Figure 5.9.

As shown in Figure 5.10, at first, the BOTC checks the data signals coming from the ATE every rising and falling edge of the clock *ate_clk_s* to determine whether the ATE has sent the instructions. If the ATE instruction has been sent for the DUT, it is first saved at the Instruction Register File (IRF) at the positive or negative edge of the clock *ate_clk_s*, and is decoded at the trailing positive or negative edge of the clock *ate_clk_sd*. The decoded values are first stored at *temporary control registers*. At the time ATE instruction has been received, an *interrupt* signal is asserted if the FSM is currently running and incoming instruction requires the operation of the FSM to be

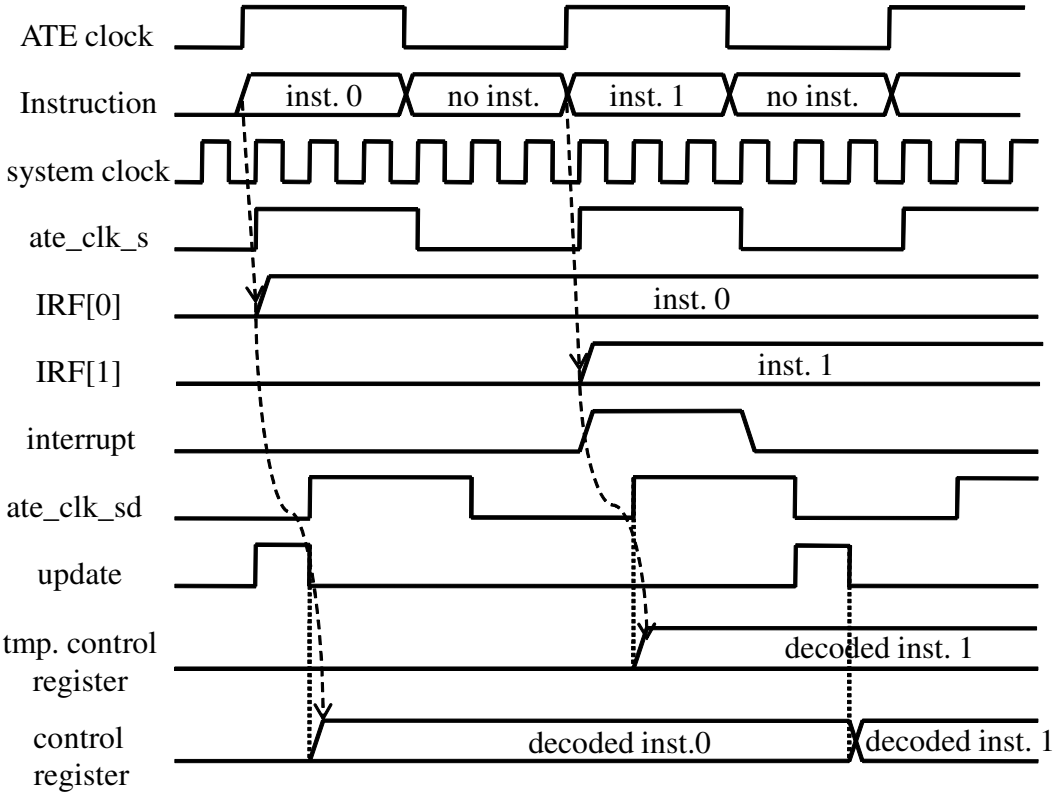


Figure 5.10: Timing Diagram of Handshake Procedure

modified (*inst.1* in Figure 5.10). Once the *interrupt* signal goes on, the FSM asserts an *update* signal one cycle before current mode of FSM operation is done, and after the *update* signal is asserted, the values of the control registers are updated with the values stored at the *temporary control registers*

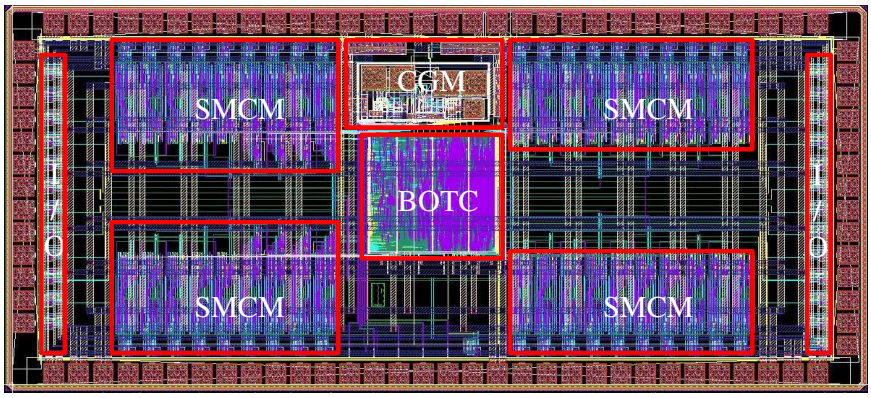
If the FSM is currently not running, or the ATE instruction does not affect the operation of the FSM, then the *update* signal is directly asserted so that the incoming instruction can be immediately decoded and updated to the control register value or to the BOTI operation (*inst.0* in Figure 5.10).

5.3 Experimental Results

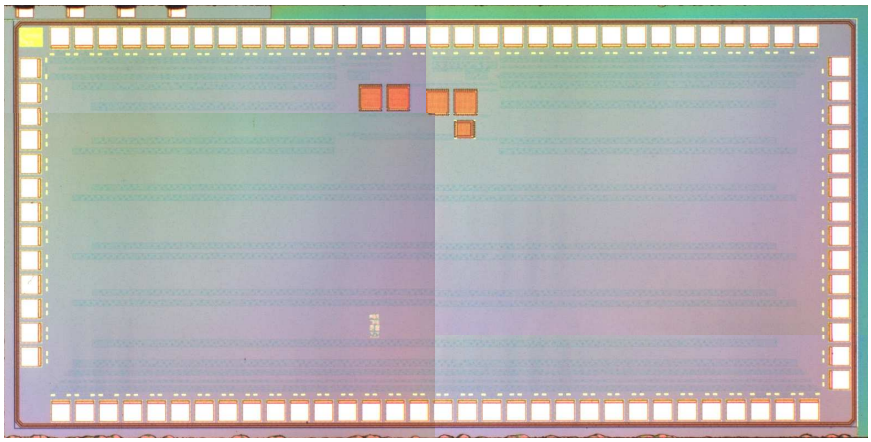
The BOTI module described in this chapter has been designed and fabricated in a 130nm CMOS technology. The BOTC has been designed using Verilog HDL and synthesized to the technology, while the CGM and the SMCM have been custom-designed. Figure 5.11 shows the layout and die photo of the BOTI chip where the BOTC, CGM and SMCM are all integrated into one chip. (In the die photo, layers of interest are not seen due to the top metal dummies). The total size of the BOTI module is $1.5mmX3.2mm$ including bond pads and ESD circuitry while the size of the BOTC module is $500\mu mX500\mu m$ and the size of the CGM is $630\mu mX370\mu m$. Also, the size of each SMCM is $300\mu mX85\mu m$ and there are 38 SMCM instantiations in the BOTI chip, so, the SMCM modules take about half of the total chip area. Current version of the BOTI chip has 26 I/O channels to communicate with the ATE and 38 I/O channels to communicate with the DUT.

In this section, post-layout simulation and post-silicon measurement results are presented to verify the proposed test framework and the operation of the fabricated BOTI chip. The case of high-speed memory test is presented in order to provide the practical example of operating complex test procedure[3].

[3]The application of the BOTI method itself can be extended to various test cases including pseudorandom test using the proper ATE instructions as explained in Section 5.1.3.



(a) Layout of the BOTI



(b) Die Photo of the BOTI

Figure 5.11: Layout and Die Photo of the BOTI

5.3.1 Post-Layout Simulation Results

The netlist of the BOTI chip has been extracted from the layout and used to run post-layout simulation with a DUT model to validate the BOTI algorithm. The Verilog HDL model of a 512Mx8, 800Mbps commercial DDR2 SDRAM module [4] was used for the simulation. The test procedure starts with setting the PLL clock multiplication ratio and locking the high-speed clocks generated by the CGM. The period of the clock signal and instructions coming from the ATE model was set to 42.5ns, so, the CGM multiplied the clock frequency by a factor of 17 to generate the system clock with the frequency of 400MHz. Figure 5.12 shows the post-layout simulation results of clock signals generated by the CGM. Rise and falls times of each clock signal are 40ps and the clock uncertainty between different clocks is less than 37.5ps at 400MHz. Also, the duty cycle of each clock signal is 0.507 in the worst case and it takes 2μs to lock the clock signals.

Once the clock signals are completely locked, the ATE sends the instructions to run the FSM at $400MHz$ which starts with setting the values of control registers. The March test requires 30 control registers in the BOTC module and most of them are 16-bit. So, it takes fifteen cycles of the ATE clock, or 637.5ns, to set the values of those registers. In this example, March C^- test is used as a test sequence which can be represented as follows.

$$\Uparrow (W0) \Uparrow (R0W1) \Uparrow (R1W0) \Downarrow (R0W1) \Downarrow (R1W0) \Downarrow (R0) \tag{5.1}$$

Once the actual March test procedure begins, the ATE model sends the instructions only when changing element operations (for example, changing the

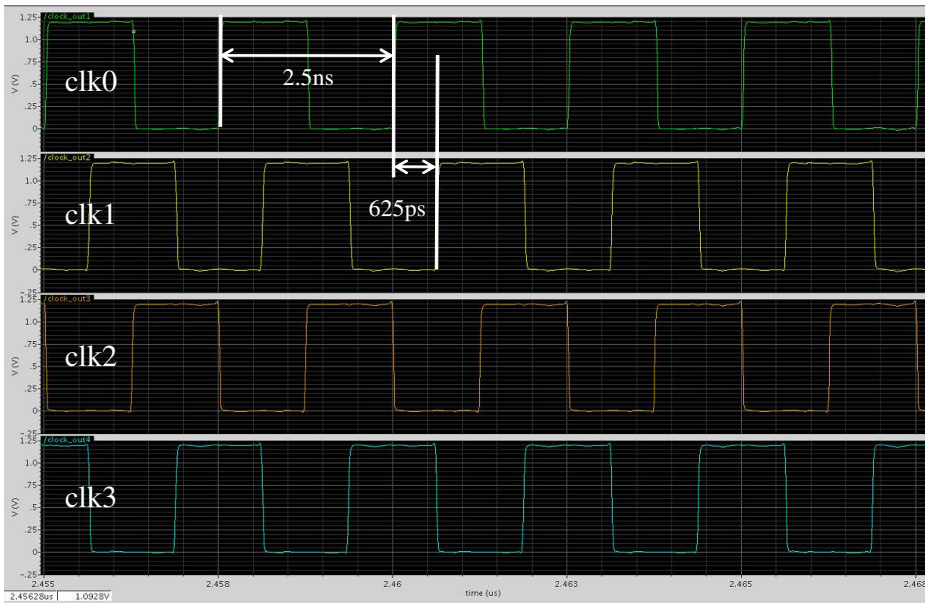
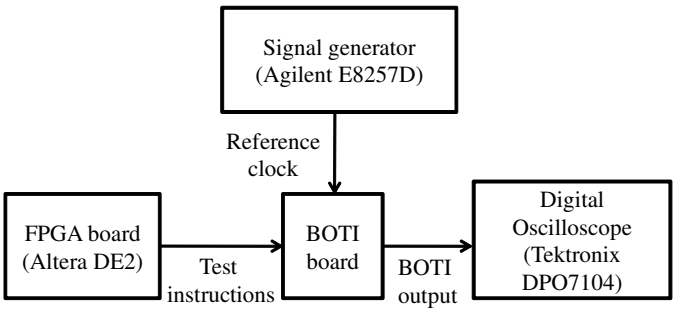
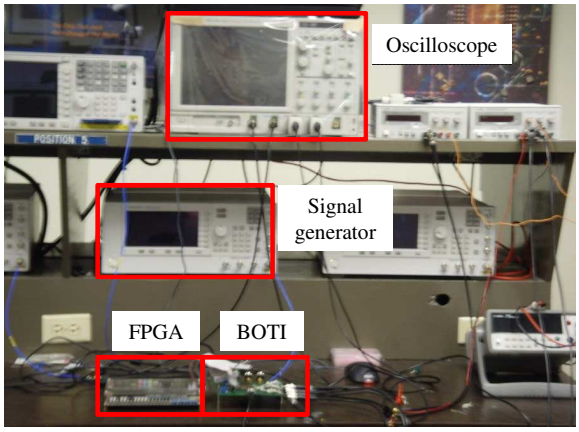


Figure 5.12: High-speed Clock Signals

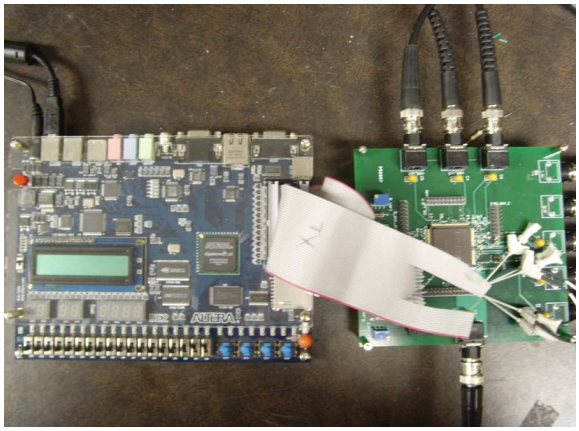
operation from $\Uparrow (W0)$ to $\Uparrow (R0W1)$) or when changing the bank address is required. These instructions modify the values of control registers which refer to *mode*, *address* and *data control*, and in this example, the ATE sends a total of 144 instructions to the BOTI for this purpose. The test procedure finishes successfully after about 281 million cycles of the system clock, or 0.7s. The time required to set the BOTI chip environment which includes CGM clock setting, skew measurement and FSM control register setting is about 3μs. Thus, the test time overhead is negligible compared to the total test time.



(a) Post-Silicon Measurement Setup



(b) Actual Measurement Environment



(c) FPGA board and BOTI board

Figure 5.13: Measurement Setup

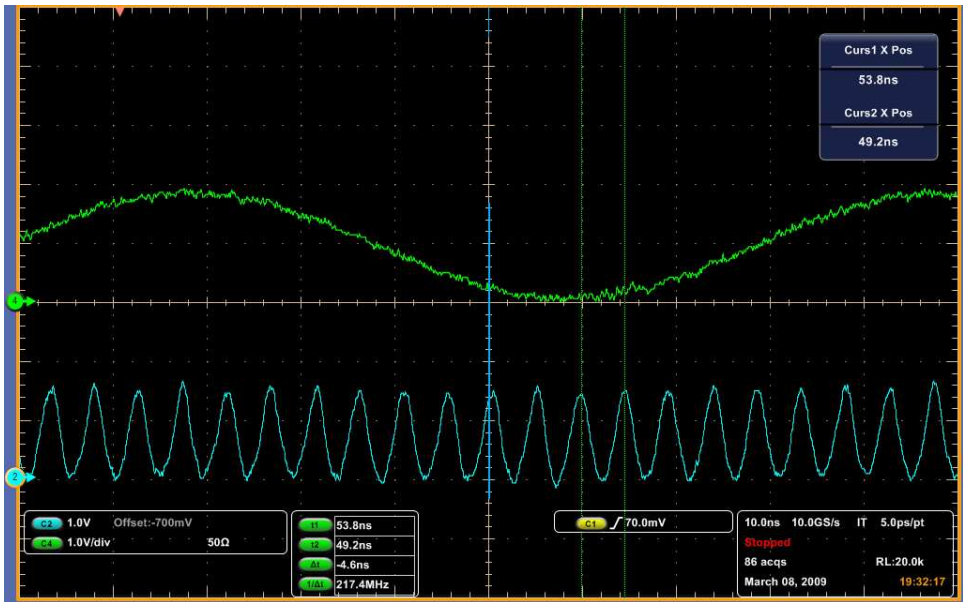
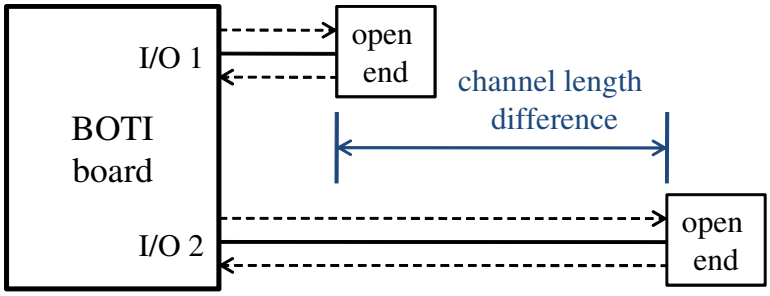


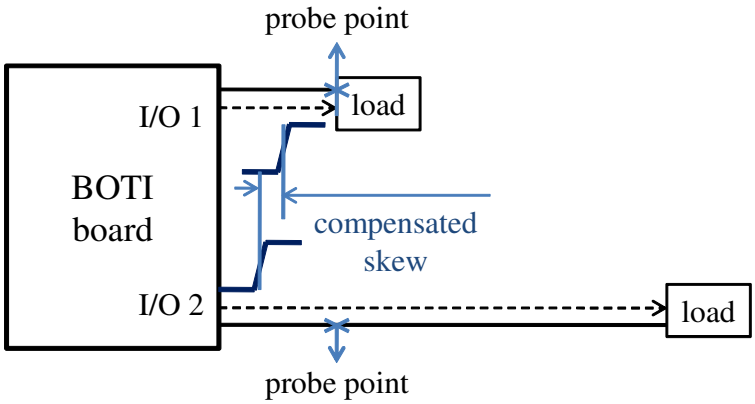
Figure 5.14: BOTI System Clock and Reference Signal

5.3.2 Post-Silicon Measurement Results

In this section, the post-silicon measurement results are presented. Figure 5.13 shows the measurement setup. The FPGA board was used to emulate the slow-running ATE and to generate the test instructions required for the operation of the BOTI. In this section, the term *BOTI board* is referred as the printed circuit board containing the BOTI. The reference signal (a sinusoidal wave is used as the reference signal) was coming from the signal generator which was not synchronized with the operation of the FPGA board. This setup is intended to introduce synchronization mismatch between the reference signal and the test instructions to see if the BOTI chip can run correctly while the reference signal is not synchronized with the test instructions. Figure 5.14



(a) Skew Measurement



(b) Skew Compensation

Figure 5.15: Skew Measurement and Compensation Setup

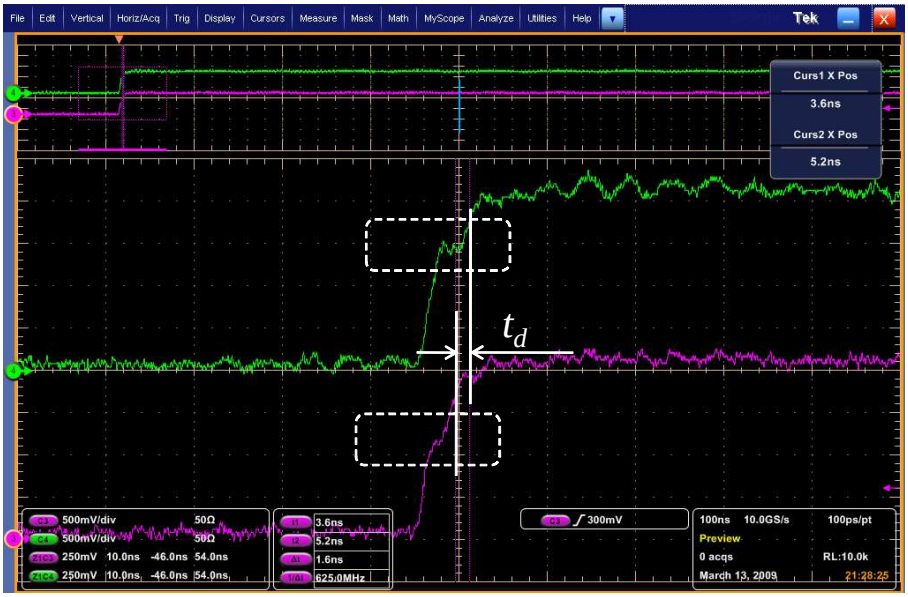
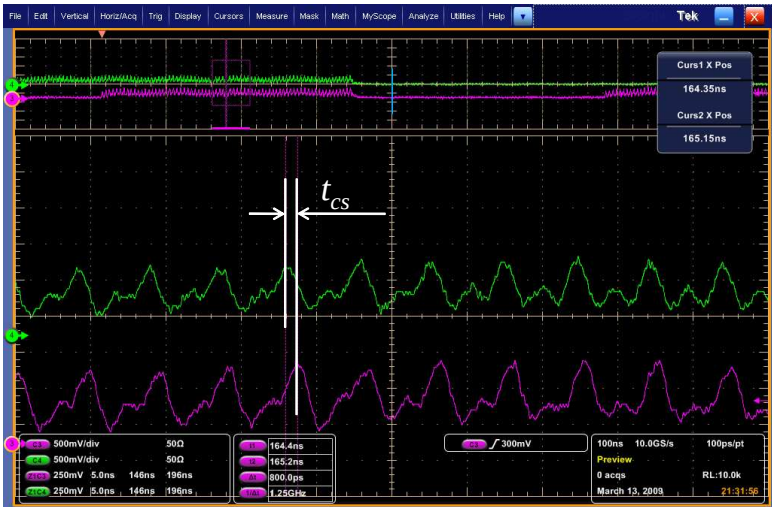


Figure 5.16: Measurement Result of Skew Calibration Signal

shows the low-frequency reference signal and the high-frequency system clock synthesized by the CGM. The reference signal shown in Figure 5.14 was running at 12.8 MHz, and the synthesized clock was running at 217.4MHz which is 17 times faster than the reference signal.

After the system clock has been synthesized, the FPGA sends the instructions to the BOTI to start measuring the off-chip channel skew using the SMCM. To verify the correct functionality of the SMCM, an intentional channel skew is introduced by making the off-chip channel length different between two I/O channels as shown in Figure 5.15(a). As described previously, in the skew measurement stage, the BOTI first sends the calibration signal to each channel and measure the time it required for the calibration signal to be



(a) Signals Measured at the Probe Point

(b) Signals Measured at the Load

Figure 5.17: Skew Compensation Results

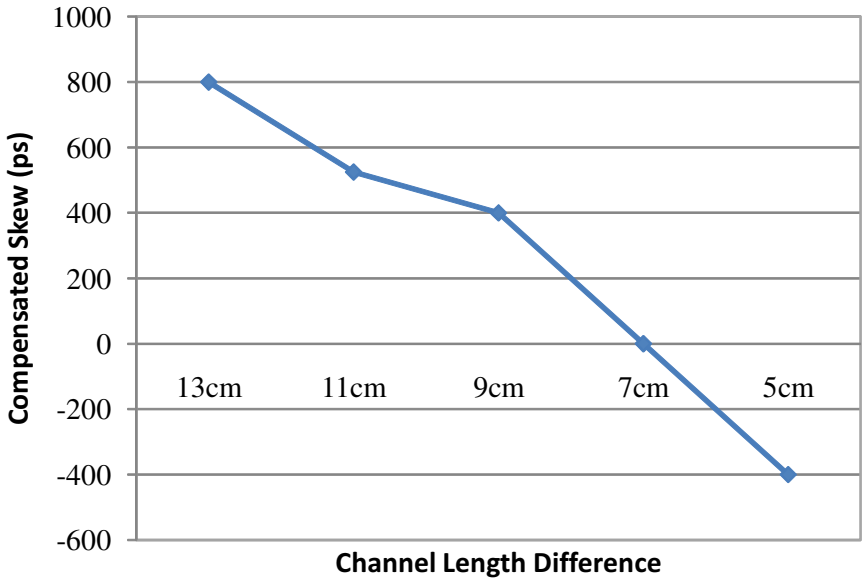


Figure 5.18: Value of Compensate Skew for Various Channel Length Differences

returned from the other end (*open end*) of the channel. Figure 5.16 shows the two calibration signals launched at two I/O channels where the upper signal was measured at the *I/O 2* and the bottom signal was measured at the *I/O 1*. Both calibration signals first rose to V_{cal} which is 0.75V (enclosed by dotted boxes) and stayed there until the calibration signals are returned from the open end of each channel. As can be seen from the figure, the signal measured at the *I/O 2* stayed at V_{cal} longer than the signal measured at the *I/O 1*, and this is due to the fact that the channel length of the *I/O 2* is longer than that of the *I/O 1*. Then, the SMCM senses the time difference t_d, which was $1.6ns$, shown in Figure 5.16 and use this information to adjust the launching time of the signals in each I/O channel to compensate for the skew. In this example,

the trigger voltage (V_{tr}) was set to 0.85V.

Next, Figure 5.17 shows the actual data signals sent from the *I/O 1* and the *I/O 2*. The signals shown in Figure 5.17(a) were measured at the *probe point* shown in Figure 5.15(b) which are located at the same distance from each I/O pin on the BOTI board, while the signals shown in Figure 5.17(b) were measured at the load of each I/O channel. The value of the compensated skew should be half the value of the t_d, since the t_d considers the time difference for round-trip signals. As we can see from those figures, the signal from the *I/O 2* was launched t_{cs}, or $800ps$ (which is half the value of t_d shown in Figure 5.16 as expected), earlier than the signal from the *I/O 1* so that two signals can arrive at the load simultaneously. Also, Figure 5.18 shows the values of compensated skew versus various channel length differences. As can be seen from the figure, the value of the compensated skew is well correlated with the channel length difference, and the maximum resolution of the skew compensation was $130ps$. (Negative value shown in Figure 5.18 was caused by the channel length difference inside the BOTI board.) This example shows that the BOTI can actually measure the off-chip skew and compensate the outgoing signals to offset the skew.

Next, the clock frequency is varied to examine whether the BOTI can generate the March test pattern according to Equation 5.1 successfully using the test instructions coming from the FPGA board. The BOTI module could generate the correct test pattern up to 220MHz without considerable signal degradation. Also, the maximum difference in operating speed between the

FPGA and the BOTI was a factor of 21. This means that, in practical case, the ATE can effectively execute a test program 21 times faster than its own speed. The main factor that limits the higher-speed operation is the long bonding wire in the package. Current version of the BOTI chip has been packaged with a 120-pin CQFP (Ceramic Quad Flatback) package. This leads to long bonding wires, which limits high-speed signal swing due to the large inductance associated with the bonding wire. This would not have been an issue if the BOTI chip had been packaged with flip-chip technology, for example. Thus, improvements in performance can be made possible by using different types of package such as flip-chip package or other types of smaller package.

In this measurement example, the BOTI operation requires 25 data channels from the FPGA board and one signal channel from the signal generator. This means that, in practical case, the ATE requires 26 I/O channels (excluding DC supply) to interface the BOTI, while it requires 38 I/O channels (excluding DC supply) to interface the DUT (which used in this example [4]) if not using the BOTI module. Thus, in this example, the number of the ATE I/O channels can be saved by 28% if the BOTI test framework is used. While the actual savings can be varied for different test cases, a possibility of saving the ATE I/O channels has been clearly shown in this example.

5.4 Summary

In this chapter, an efficient at-speed test framework for high-speed DUTs using low-speed ATE has been discussed. The framework presented

in this chapter uses a custom-designed built-off test interface (BOTI) circuit to generate at-speed test patterns for a high-speed DUT, and to analyze test responses coming from the DUT to determine whether the DUT is faulty. Since the ATE actively controls the operation of the BOTI and overall test procedure, the test program run on the BOTI module is highly programmable, and thus the presented BOTI test method can be flexibly applied to various types of DUT. Moreover, the issues arising from implementing the test interface circuit off-chip has been addressed in this chapter. The BOTI measures off-chip skews among different I/O channels and compensate them for the signals communicating with the DUT. Using the presented BOTI method, test cost can be decreased considerably by extending the use of low-speed ATE to at-speed test of higher-speed DUT.

The concept of the BOTI module has been implemented in a semiconductor chip using 130nm technology and the experimental results show the correct functionality of the BOTI chip and the validity of the proposed test framework which uses the BOTI chip.

Chapter 6

Conclusion

This dissertation presents novel functional test methods for analog and mixed-signal circuits which focused on characterizing the performances of non-linear devices using a low-cost test environment. The major contribution of the work presented in this dissertation is to develop efficient methods to measure the conventional specification parameters of AMS circuits using non-conventional and cost-effective test methods. This can be possible by abstracting the DUT behaviors accurately using system level performance modeling and by developing performance characterization methods that use easy-to-generate test stimuli and low-complexity characterization algorithms. In doing so, we can bridge the gap between lowering the test cost/test time and maintaining good test accuracy, which has been the important issue in the area of AMS circuit test. The methods presented in this dissertation can be divided into two categories. The first is to devise a new test framework for AMS circuits that is aimed at using a simple test environment without compromising the test accuracy. The second is to reduce the test time by testing multiple DUTs simultaneously while not affecting the test accuracy of an individual DUT.

The low-cost functional test method presented in Chapter 3 uses a pseudorandom signal to measure the dynamic performance parameters of AMS circuits. Unlike the previous alternate functional test methods which calculate the performance parameters indirectly using alternate test signatures, the pseudorandom test method calculates the performance parameters directly. The pseudorandom signal is chosen as a test stimulus because it contains multiple sinusoidal tones which cover a wide range of frequencies, and thus can be used to extract the spectral information of the DUT efficiently. More importantly, the pseudorandom signal can be generated using an uncomplicated test equipment or on-chip LFSR which helps to reduce the test cost considerably. Also, by using the LFSR in generating the test stimulus, we can relax the difficulty in accessing the embedded AMS circuits from external test equipment to deliver the test stimulus to the DUT. The Volterra series model of the DUT is used to analyze the test response of the nonlinear devices and to develop a low-complexity characterization algorithm to extract the required information from the test response. The pseudorandom test method developed in this research was verified using actual hardware measurements to calculate the dynamic performance parameters of a commercial mixed-signal IC. The measurement results show that the method can measure various specification parameters with a small amount of error compared to the conventional test methods.

The parallel test algorithm featured in Chapter 4 shows methods to reduce the test time by increasing the test throughput beyond the level limited

by tester resources. Unlike the previous parallel test methods in which the parallelism is extended by increasing the performance of the tester resources, the parallel test methods increase the parallelism by exploring the spectral characteristics of the test stimulus and the system level description of the DUT. Thus, the new methods developed can be implemented with existing test equipment without incurring an additional test cost while reducing the test time considerably. In these methods, multiple DUTs share common test equipment and the test response from one DUT is mixed with the test responses from other DUTs. It was shown in Chapter 4 that we can identify the performance parameters of each DUT from the composite test responses using the system level performance characterization methods with the help of simple circuits such as an analog adder and an RMS power detector placed on the signal path. Various experimental results presented in Chapter 4 demonstrate that the presented parallel test methods can be actually used to characterize the performance of individual DUTs separately which are tested in parallel.

The goal of the built-off test interface method presented in this dissertation is to relax the cost associated with the tester resources in the aspect of tester speed. The low-cost functional test methods described in this dissertation are aimed at reducing the test cost by using low-cost test stimuli and at reducing the test time by increasing the test throughput. One issue with these methods, though, is that they still require at-speed test patterns to excite the DUT. The BOTI module described in Chapter 6 can be used to generate high-speed test stimuli and to capture the test responses accurately under the

control of a low-speed ATE. Thus, the presented BOTI method makes it possible to test high-speed DUTs using lower-speed testers with the help of BOTI module. This method was validated through hardware measurements using a fabricated BOTI chip, and the results show that, in a practical case, we can control the at-speed test of high-speed DUT using low-speed test equipment.

Overall, this dissertation presents both theoretical and practical studies developing the new functional test methods for AMS circuits to solve the various problems associated with the conventional test method. The results described in this dissertation can be extended to further research in the area of AMS and RF circuit test. Some of the possible future directions can be listed as follows.

- The pseudorandom test algorithm presented in this dissertation is aimed at finding dynamic performance parameters of AMS circuits. The algorithm developed in this dissertation can be further exploited to characterize static performance parameters such as integrated-nonlinearity (INL), differential-nonlinearity (DNL), etc. Traditionally, the static parameters of AMS circuits are characterized using a DC signal or a ramp signal. By exploring the spectral relationship between the pseudorandom signal and the conventional static test stimuli, we can use the pseudorandom signal to characterize the static parameters, and thus extend the application of the pseudorandom test further.

- In the area of AMS circuit test, one of the important trends in recent

years is to maximize the use of digital test equipment in testing analog functionality. The parallel loopback test method presented in Chapter 4 shows the possibility of configuring a digital-in/digital-out environment and using digital test equipments to test AMS circuits. We can implement the similar loopback test environment for the parallel pseudorandom test method explained in this dissertation. This will require research on understanding the characteristics of pseudorandom signals in a multi-stage nonlinear system. Upon implementing the loopback environment for pseudorandom test, we can use this method for self-test of embedded AMS circuits with the help of on-chip LFSR and DSP which are common components in the current SOC devices.

- The functionalities of the BOTI module explained in Chapter 5 can be further extended to handle more complicated test signals including analog signals. This can be done by improving the FSM structure to increase its degrees of freedom so that it can generate various test patterns flexibly. To cope with analog signals, existing on-chip test stimulus generation techniques can be studied along with the BOTI structure to generate and to capture the analog signals accurately.

Bibliography

[1] 12-bit TxDAC^R D/A Converter Data Sheet, http://www.analog.com.

[2] Advantest^{TM} Mixed-Signal Test System, http://www.advantest.com.

[3] *International Technology Roadmap for Semiconductors*, 2006 update, http://www.itrs.net.

[4] Samsung^{TM} DRAM Simulation Model, http://www.samsung.com.

[5] TruPwr^{TM} Detector Data Sheet, http://www.analog.com.

[6] S. Akbay, A. Halder, A. Chatterjee, and D. Keezer. "Low-Cost Test of Embedded RFAnalogMixed-Signal Circuits in SOPs". *IEEE Trans. Advanced Packaging*, vol. 27, no. 2, pp. 352–363, 2004.

[7] S.S. Akbay and A. Chatterjee. "Feature Extraction Based Built-In Alternate Test Of Rf Components Using a Noise Reference". *proc. VLSI Test Symposium*, 2004.

[8] K. Arabi and B. Kaminska. "Oscillation-Test Methodology for Low-Cost Testing of Active Analog Filters". *IEEE Trans. Instrum. Meas.*, vol. 49, no. 4, pp. 798–806, 1999.

[9] H. Bogard and C. Repasky. "Increasing Test Throughput Through The Implementation Of Paralle Test On A 16-Bit Multimedia Audio Codec". *Proc. International Test Conference*, pp. 370–376, 1995.

[10] D. Brown, J. Ferrario, R. Wolf, J. Li, and J. Bhagat. "RF Testing On a Mixed Signal Tester". *Proc. International Test Conference*, pp 793–800, 2004.

[11] M. Burns and G.W. Roberts. *An Introduction to Mixed-Signal IC Test and Measurement*. McGRAW Hill, 2001.

[12] J.H. Chun, H. Yu, and J.A. Abraham. "An Efficient Linearity Test for On-chip High-Speed ADC and DAC Using Loop-back". *proc. ACM Great Lakes Symposium on VLSI*, pp. 328–331, 2004.

[13] J. Dabrowski and J.G. Bayon. "Mixed Loopback Bist for Rf Digital Transceivers". *Proc. International Symposium on Defect and Fault Tolerance in VLSI Systems*, pp. 220–228, 2004.

[14] B. Dufort and G.W. Roberts. "On-Chip Analog Signal Generator For Mixed-Signal Built-In Self-Test". *Proc. Custom Integrated Circuit Conference*, pp. 549–552, 1998.

[15] M.G.C. Flores, M. Negreiros, L. Carro, and A.A. Susin. "Inl and Dnl Estimation Based on Noise for Adc Test". *IEEE Trans. Iinstrument and Measurement*, vol. 53, no. 5, pp. 1391–1395, 2004.

[16] F.R. Gantmacher. *The Theory of Matrices, Vol.2*. AMS Chelsea, 2000.

[17] M.M. Hafed and G.W. Roberts. "A Stand-Alone Integrated Excitation/Extraction System for Analog BIST Applications". *Proc. Custom Integrated Circuit Conference*, pp. 83–86, 2000.

[18] K. Han, J. Park, J. Lee, J.A. Abraham, E.Byun, C. Woo, and S. Oh. "Low-Complexity Off-Chip Skew Measurement and Compensation Module Design for Built-off Test Chip". *Proc. European Test Symposium*, 2009.

[19] M.O.J. Hawksford. "System Measurement and Modeling Using Pseudo-random Filtered Noise and Music Sequences". *114th AES Convension.*, Amsterdam, Mar. 2003.

[20] G. Huertas, D. Vazquez, E.J. Peralias, A. Rueda, and J.L. Huertas. "Practical Oscillation-Based Test of Integrated Filters". *IEEE Trans. Des. Test Comput.*, vol. 19, pp. 64–72, 2002.

[21] L. Jin, H. Haggag, R. Geiger, and D. Chen. "Testing of Precision DACs using Low-Resolution ADCs with Dithering". *Proc. International Test Conference*, pp. 1–10, 2006.

[22] P. Kabisatpathy, A. Barua, and S. Sinha. "Artificial Neural-Network Model-Based Observers". *IEEE Circuits and Devices Magazine*, pp. 18–26, July/August, 2005.

[23] V. Kerzerho, P. Cauvet, S. Bernard, F. Azais, M. Comte, and M. Renovell. "A Novel Dft Technique for Testing Complete Sets of Adcs and Dacs in Complex Sips". *IEEE Trans. Design and Test of Computers*, vol. 23, no. 3, pp. 234–243, 2006.

[24] B. Kim, Z. Fu, and J.A. Abraham. "Transformer-Coupled Loopback Test for Differential Mixed-Signal Specifications". *Proc. VLSI Test Symposium*, pp. 291–296, 2007.

[25] B. Kim, H. Shin, J.H. Chun, and J.A. Abraham. "Optimized Signature-Based Statistical Alternate Test for Mixed-Signal Performance Parameters". *Proc. European Test Symposium*, pp. 199–204, 2006.

[26] J. Kwan and Q. Zhao. "A Novel Ganged Dac Solution for Multi-Site Testing". *Proc. International Test Conference*, pp. 1–6, 2006.

[27] V.Z. Marmarelis. *Nonlinear Dynamic Modeling of Physiological Systems.* Wiley, 2004.

[28] J.M. Mendel. "Tutorial on Higher-Order Statistics (Spectra) in Signal Processing and System Theory: Theoretical Results and Some Applications". *Proc. IEEE*, vol. 79, no. 3, pp. 278–305, 1991.

[29] P. Mikulik and J. Saliga. "Voltera Filtering for Integrating Adc Error correction, Based on A Priori Error Model". *IEEE Trans. Iinstrument and Measurement*, vol. 51, no. 4, pp. 870–875, 2002.

[30] L.S. Milor. "A Tutorial Introduction to Research on Analog and Mixed-signal Circuit Testing". *IEEE Trans. Circuits and Systems II*, vol. 45, no. 10, pp. 234–243, 1998.

[31] V. Natarajan, G. Srinivasan, A. Chatterjee, and C. Force. "Novel Cross-Loopback Based Test Approach for Specification Test of Multi-Band,

Multi-Hardware Radios". *Proc. VLSI Test Symposium*, pp. 297–302, 2007.

[32] M. Negreiros and A.A. Susin. "Low Cost Analog Testing of Rf Signal Paths". *proc. Design Automation Test Eur.*, 2004.

[33] M. Nelms, K. Gorman, and D. Anand. "Generating At-Speed Array Fail Maps with Low-speed ATE". *Proc. VLSI Test Symposium*, pp. 87–92, 2004.

[34] R.D. Nowak and B.D. Van Veen. "Random and Pseudorandom Inputs for Volterra Filter Identification". *IEEE Trans. Signal Processing*, vol. 42, no. 8, pp. 2124–2135, 1994.

[35] C.Y. Pan and K.T. Cheng. "Pseudorandom Testing for Mixed-Signal Circuits". *IEEE Trans. Computer-Aided Design*, vol. 16, no. 10, pp. 1173–1185, 1994.

[36] J. Park, J. Chung, and J.A. Abraham. "LFSR-based Performance Characterization of Analog and Mixed-Signal Circuits". *Proc. Asian Test Symposium*, 2009.

[37] J. Park, S. Madhavapeddi, A. Paglieri, C. Barr, and J.A. Abraham. "Defect-based Analog Fault Coverage Analysis using Mixed-mode Fault Simulation". *Proc. International Mixed-signals, Sensors, and Systems Test Workshop*, 2009.

[38] J. Park, H. Shin, and J.A. Abraham. "Pseudorandom Test for Nonlinear Circuits Based on a Simplified Volterra Series Model". *Proc. International Symposium on Quality Electronic Design*, pp. 495–500, 2007.

[39] J. Park, H. Shin, and J.A. Abraham. "Parallel Loopback Test of Mixed-Signal Circuits". *Proc. VLSI Test Symposium*, pp. 309–316, 2008.

[40] J.G. Proakis and D.G. Manolakis. *Digital Signal Processing.* Prentice Hall, 1996.

[41] J.G. Proakis and M. Salehi. *Communication Systems Engineering.* Prentice Hall, 2002.

[42] B. Provost and E. Sanchez-Sinencio. "On-Chip Ramp Generators for Mixed-Signal BIST and ADC Self-Test". *IEEE J. Solid-State Circuits*, vol. 38, no. 2, pp. 1821–1834, 2003.

[43] A. Raghunathan, H. Shin, J.A. Abraham, and A. Chatterjee. "Prediction of Analog Performance Parameters Using Oscillation Based Test". *proc. VLSI Test Symposium*, pp. 377–382, 2004.

[44] B. Razavi. *Design of Analog CMOS Integrated Circuits.* McGRAW Hill, 2001.

[45] B. Razavi. *RF Microelectronics.* McGRAW Hill, 2001.

[46] S. Sasho and M. Shibata. "Multi-Output One-Digitizer Measurement". *Proc. International Test Conference*, pp. 258–264, 1998.

[47] H. Shin, B. Kim, and J.A. Abraham. "Spectral Prediction for Specification-Based Loopback Test of Embedded Mixed-Signal Circuits". *proc. VLSI Test Symposium*, pp. 412–419, 2006.

[48] H. Shin, J. Park, and J.A. Abraham. "A Statistical Digital Equalizer for Loopback-based Linearity Test of Data Converters". *proc. Asian Test Symposium*, pp. 245–250, 2006.

[49] H. Shin, J. Park, and J.A. Abraham. "Built-in Fault Diagnosis for Tunable Analog Systems Using an Ensemble Method". *Proc. International Test Conference*, 2006.

[50] H. Shin, H. Yu, and J.A. Abraham. "LFSR-based BIST for Analog Circuits Using Slope Detection". *proc. ACM Great Lakes Symposium on VLSI*, pp. 316–321, 2004.

[51] H. Stark and J.W. Woods. *Probability and Random Processes with Applications to Signal Processing.* Prentice Hall, 2002.

[52] A. Valdes-Garcia, R. Venkatasubramanian, R. Srinivasan, J. Silva-Martinez, and E. Sanchez-Sinencio. "A CMOS RF RMS Detector for Built-in Testing of Wireless Transceivers". *Proc. VLSI Test Symposium*, pp. 249–254, 2005.

[53] P.N. Variyam, S. Cherubal, and A. Chatterjee. "Prediction of Analog Performance Parameters Using Fast Transient Testing". *IEEE Trans. Computer-Aided Design*, vol. 21, no. 3, pp. 349–361, 2002.

[54] E.H. Volkerink, A. Khoche, J. Rivoir, and K.D. Hilliges. "Test Economics for Multi-site Test with Modern Cost Reduction Techniques". *Proc. VLSI Test Symposium*, pp. 411–416, 2002.

[55] R.H. Walden. "Analog-to-Digital Converter Survey and Analysis". *IEEE Journal on Selected Areas in Communicatoin*, vol. 17, no. 4, pp. 539–550, 1999.

[56] P. Wambacq, P. Dobrovolny, S. Donnay, M. Engels, and I. Bolsens. "Compact Modeling of Nonlinear Distortion in Analog Communication Circuits". *proc. Design Automation Test Eur.*, pp. 350–354, 2000.

[57] P. Wambacq and W.M. Sansen. *Distortion Analysis of Analog Integrated Circuits*. Kluwer Academic Publishers, 1998.

[58] L-T. Wang, C-W. Wu, and X. Wen. *VLSI Test Principles And Architectures*. Morgan Kaufmann, 2006.

[59] H. Yu, J.A. Abraham, S. Hwang, and J. Roh. "Efficient Loop-back Testing of On-chip ADCs and DACs". *Proc. Asia and South Pacific Design Automation Conference*, pp. 651–656, 2003.

[60] H. Yu, H. Shin, J.H. Chun, and J.A. Abraham. "Performance Characterization of Mixed-Signal Circuits Using a Ternary Sinal Representation". *Proc. International Test Conference*, pp. 1389–1397, 2004.

[61] C. Zhang, R. Gharpurey, and J.A. Abraham. "Low Cost RF Receiver Parameter Measurement with On-Chip Amplitude Detectors". *Proc. VLSI Test Symposium*, pp. 203–208, 2008.

Vita

Joonsung Park was born in Seoul, Korea, the son of Ki-ho Park and Jung-ok Kang. He received the Bachelor of Engineering degree in Electrical Engineering from the Korea University, Seoul, Korea in 2003, and the Master of Science degree in Electrical Engineering from the University of Michigan, Ann Arbor, MI, USA in 2005. He joined the Computer Engineering program in August 2005 at the University of Texas at Austin to pursue a Ph.D. degree. His area of research has been analog and mixed-signal circuit test methodology. He is currently working at Texas Instruments, Inc., Dallas, TX in wireless analog and RF group.

Permanent address: Hyundai Mansion 1-102, Bundang,
Gyunggi, South Korea 463-030

This dissertation was typeset with LaTeX[†] by the author.

[†]LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.